VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Jednoduše vyplnitelné PDF a online formuláře

Simple Way to Fill PDF and Online Forms

2010 Nevřala Zdenko

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostravě 30. dubna 2010 .....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2010 .....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli. Jmenovitě pak panu Ing. Ladislavu Martinčíkovi za pečlivé a odborné vedení práce. Dále bych chtěl poděkovat svým rodičům za stálou podporu během studia.

Abstrakt

Diplomová práce se zabývá tématikou PDF formulářů. Cílem je vytvořit PDF portál, umožňující uživatelům vytvářet a definovat vlastní šablony z již existujících PDF formulářů, a z těchto šablon generovat dokumenty s výstupem do PDF. Uživatel bude moci vytvářet interaktivní prvky v šablonách, které mohou být součástí kalkulace, nebo budou validovány pomocí validací. V práci se seznámíme s technologiemi použitými pro vývoj (např. CouchDB), se způsobem a nástroji pro testování (RSpec, Cucumber). Dále je v práci zahrnutý návrh, analýza a implementace portálu. Implementaci jsem se rozhodl řešit v jazyce Ruby za použití frameworku Ruby on Rails (RoR). Práce také obsahuje programátorskou, uživatelskou a administrátorskou příručku.

Klíčová slova: PDF portál, PDF formulář, CouchDB, Rspec, Cucumber, Ruby on Rails

Abstract

Diploma thesis is concentrating on PDF forms. The goal is to create PDF portal, letting people create and define custom templates from already existing PDF forms, and from this templates generate documents with output to PDF. User will be able to create interactive elements in templates, which can be part of the calculation or will be validated. In this thesis we will get know used technologies for developing (ex. CouchDB), with methodology and tools used for testing (RSpec, Cucumber). Also thesis includes design, analysis and implementation of portal. Implementation is done in Ruby and used framework is Ruby on Rails (aka RoR). Thesis contains programmers, users and administrator's guideline.

Keywords: PDF portal, PDF forms, CouchDB, RSpec, Cucumber, Ruby on Rails

Seznam použitých zkratek a symbolů

AJAX	–	Asynchronous JavaScript and XML
API	–	Application Programming Interface
BDD	–	Behavior Driven Development
CSS	–	Cascading Style Sheets
DB	–	Databáze
FDF	–	Forms Data Format
HTML	–	HyperText Markup Language
HTTP	–	HyperText Transfer Protocol
JSON	–	JavaScript Object Notation
MVC	–	Model View Controller
PDF	–	Portable Document Format
REST	–	Representational State Transfer
RoR	–	Ruby on Rails
STDOUT	–	Standart Output
TDD	–	Test Driven Development
URL	–	Uniform Resource Locator
UUID	–	Universally Unique Identifier
WWW	–	World Wide Web
XHTML	–	Extensible Hypertext Markup Language
XML	–	Extensible Markup Language

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Cílem diplomové práce je vytvořit PDF portál, který by uživateli umožnil pracovat s existujícími PDF formuláři. Registrovaný uživatel zde bude moci nahrávat na svůj účet existující PDF formuláře (dále jen formuláře). K nim bude moci vytvářet své šablony dle vlastních potřeb. To znamená, že bude vytvářet vlastní vzhled šablony, kalkulace, prvky, které se mají v šabloně vyskytovat a určovat jejich výchozí hodnoty a validace. Kalkulace se dají využít pro výpočet finančních či matematických vzorců, které se například využívají pro výpočet daňového přiznání. Systém by měl také poskytovat veřejný repozitář[1], z kterého (popř. do kterého) si mohou uživatelé nahrávat formuláře a šablony na svůj vlastní účet. Formuláře i šablony budou ve veřejném repozitáři kategorizovány pomocí značek (tagů) jako: daňe, plná moc ... Pomocí šablon uživatel vytváří vyplněné dokumenty, které mají výstup v PDF formátu. K systému bude mít přístup administrátor, který jej bude spravovat.

PDF portál je implementován pomocí jazyka Ruby a webového frameworku Ruby on Rails. Pro práci s PDF formuláři jsem použil knihovnu „iText“, napsanou v Java jazyce. Z toho důvodu jsem musel vhodně zvolit workflow a technologii pro práci s knihovnou (viz kapitola 4). Mimo MySQL databázi jsem pro uchovávání dokumentů zvolil databázi CouchDB, která je dokumentově orientována (viz kapitola 3).

Kapitola 2 pojednává o PDF formulářích, jejich vlastnostech, výhodách a nástrojích pro práci s nimi.

V kapitole 3 popisuji databázi CouchDB, její rysy, výhody, vlastnosti a práci s ní.

Kapitola 4 nás seznámí s použitým workflow pro PDF portál, který je jednoduše rozdělen na dvě části Ruby a Java.

V Kapitole 5 je popsán návrh PDF portálu. Nalezneme zde popis tabulek, dokumentů a jejich atributů. Dále schéma databáze MySQL a CouchDB, a také případy užití portálu.

V Kapitola 6 se zabývám návrhem implementace. Zde jsem řešil problematiku technického a programového vybavení, otázku grafického vzhledu portálu a návrhy jiných programových částí portálu jako např. gramatiku pro kalkulace atd.

Kapitola 7 je samotná implementace PDF portálu. Jsou zde podrobné informace o implementaci, použité programové prostředky, umístění stránek na internetu, grafický design stránek.

Otázce testování jsem věnoval kapitolu 8. Zde se zabývám způsobem testování a technologiemi, které jsem použil.

[1] repozitář - schránka, úschovna

1.1 Stanovení cíle diplomové práce

Na základě požadavků specifikace diplomové práce na téma portál pro PDF formuláře a znalostí získaných studiem byl stanoven cíl této diplomové práce.

- seznámení s problematikou PDF formulářů
- seznámení s dostupnými technologiemi pro práci s PDF formuláři a dokumenty
- návrh portálu
- návrh funkčního workflow vzhledem k dostupným technologiím a knihovnám
- implementace portálu
- uplná nebo alespoň částečná implementace testů pro portál
- tvorba dokumentace: programátorská, administrátorská a uživatelská příručka

2 PDF Formuláře

2.1 Stručně o PDF

PDF je anglická zkratka pro Portable Document Format, což znamená „přenosný formát dokumentů“. Formát byl vyvinut firmou Adobe pro ukládání dokumentů nezávisle na softwaru či hardwaru, na kterém byly pořízeny. PDF má v dnešní době podporu nejen pro vkládání textu, ale i obrázků, videí, zvuků atd., přičemž zajišťuje, že se dokument zobrazí na všech zařízeních stejně. Pro vytváření PDF dokumentů můžeme použít buďto Adobe od Acrobatu, nebo jiný program, který z větší části podporuje pouze export do PDF. Prohlížení PDF dokumentů je podstatně snazší, protože existuje nespočetné množství prohlížečů pro různé platformy. Nejznámějším prohlížečem je oficiální prohlížeč od mateřské firmy Adobe Reader. Celkově lze říci, že je tento formát velmi rozšířený a hojně využívaný. 1.července 2008 byl publikován jako standard ISO 32000-1:2008

2.2 PDF formuláře

Každá společnost v dnešní době potřebuje ke svému chodu nejrůznější formuláře a dokumenty, které obsahují určité části pro vyplnění požadovaných údajů. Všeobecně jedny z nejvíce využívaných formulářů jsou například daňová přiznání, plné moci, kupní smlouvy atd. Kromě těchto formulářů se můžeme často setkat s formuláři typu faktura, dotazník, objednávka, přihláška, veřejný průzkum a další.

S vývojem internetu a webových aplikací, převzaly formuláře elektronickou podobu ve formátu HTML, čímž odpadly časové nároky na ruční vyplňování a zpracovávání v papírovém formátu. Později do této oblasti vstoupila i firma Adobe a umožnila udělat z jednoduchých PDF dokumentů interaktivní formuláře. Firma Adobe rozšířila PDF specifikaci o interaktivní pole a tlačítka pro získávání a zpracovávání uživatelských údajů. Další důležitou schopností nynějšího PDF je využívání vlastního klonu jazyka JavaScript, což umožňuje vytvářet ve formulářích programové procedury, díky kterým jsou interaktivnější a schopny automatizovat řadu činností. Velkou výhodou použití JavaScriptového jazyka je vytváření validací, závislostí a různých kalkulací přiřazených k jednotlivým formulářovým prvkům. Můžeme tak určit, zda hodnota form. prvku má být číslo, nebo se má vypočítat pomocí požadované kalkulace z jiných prvků.

2.3 FDF formát

Forms Data Format(FDF) je formát pro zpracování PDF formulářů. Obecná představa o FDF je podobná HTML formulářům. Rozdíl je v podstatě ve formátu, v němž se data odesílají na server. Je možné jej využívat při odesílání dat na server, pro odesílání odpovědí ze serveru a začleňování do interaktivního formuláře. Obvykle se používá pro export formulářových dat v samostatném souboru, který obsahuje jenom informace o vyplněném formuláři (jednoduše řečeno pole s hodnotami). Tento soubor se přenese na server, na kterém se vyskytuje také PDF formulář k danému FDF souboru. Pokud si uživatel přeje vidět vyplněná data, PDF formulář si načte údaje definované v daném

FDF souboru. V mé práci jsem použil vlastní zjednodušenou obdobu FDF (hash pole[2] „klíč“=>hodnota v dokumentu Document), tento formát zmiňuji pro lepší pochopení funkčnosti PDF formulářů dále.

2.4 Jak formuláře v PDF fungují

Tvůrce vytvoří PDF dokument s již zmiňovanými interaktivními formulářovými prvky, případně obohacenými o potřebnou validaci či kalkulaci. Poté dukument zveřejní. Uživatel si dokument může stáhnout do svého počítače. Po otevření dokumentu (je zapotřebí mít nainstalovaný PDF prohlížeč nebo plugin pro otevírání PDF dokumentu v internetovém prohlížeči) uživatel vyplní požadovaná data. Poté si může dokument uložit. V případě, že je formulář vytvořen k odesílání dat, může jej odeslat pomocí tlačítka „Submit“ ke zpracování. Způsob odeslání dat i místo určení jsou nastaveny od tvůrce PDF. Nejčastějším formátem dat k odeslání je formát FDF viz odstavec 2.3. Místo určení pak obvykle bývá internetový server, na kterém běží skript pro zpracování dat odeslaných z formuláře. Uživateli se dále zobrazí zpětná odezva, která může buďto potvrzovat přijetí dat, nebo dynamicky změnit vyplňovaný formulář. Bližší popis zpracovávání PDF formulářů naleznete v článku [1].

2.5 PDF vs. HTML

Výhody a nevýhody PDF formulářů jsou velmi diskutovanou a spornou otázkou. V následujicích bodech se pokusím shrnout největší výhody PDF formulářů ve srovnání s jejich rivalem HTML formulářů. Zde bych se chtěl odkázat na práci pana Marka Bobera, který pojednává o využití interaktivních formulářů PDF v e-vzdělávání [2]. Další pěkný článek o možnostech a výhodách PDF formulářů naleznete na [3].

Výhodou obou formátů je jejich elektronická povaha, která umožňuje snazší vyplňování dat, jejich zpracovávání a automatizaci.

Výhody PDF:

Vizuální věrnost - jak jsem již zmínil v odstavci 2.1, základní výhodou PDF je stejné zobrazení na všech zařízeních, což u HTML formulářů bývá mnohdy jiné díky rozdílnosti prohlížečů.

Bezpečnost - jednou z možností spjatých s PDF je zabezpečení dokumentů na mnoha úrovních, například digitální podpis, který není složité nastavit. U HTML formulářů je třeba složitěji konfigurovat server i správu s vlastnictvím klíčů nebo certifikátů.

Offline režim - PDF formuláře lze uložit bez jakéhokoliv exportu. Užival si jej jednoduše uloží na disk v originální podobě. Formuláře umožňují vyplnění a uložení dat bez připojení k internetu. Při ukládání HTML nastanou problémy s obrázky, styly apod.

[2] hash pole - asociativní pole

Tisk - Tisk PDF formulářů je velmi jednoduchý a spolehlivý, dokument vytiskne přesně tak, jak jej vidíme v elektronické podobě. U HTML formulářů je to složitější. Pokud dáme tisknout formulář, jehož stránka nemá specifikovaný CSS styl pro tisk, vytiskne se nám celá HTML stránka tak, jak ji vidíme - s případným menu, bannery atd. U stránek, které mají nadefinovaný CSS styl pro tisk, se formulář mnohdy nepodobá tomu, jak jej vidíme v elektronické podobě.

Nevýhody PDF:

Mobilní technologie - většina webových aplikací se zaměřuje i na rozhraní pro mobily. Tady má HTML výhodu, neboť se může uživateli v mobilním prohlížeči zobrazit příznivěji vzhledem k velikosti mobilního displeje. PDF se v mobilech stále zobrazuje v originální velikosti a podobě, což bývá neúměrné k velikosti zobrazovacího zařízení mobilního přístroje.

Cena - Velikou nevýhodou PDF formulářů, kterou bych chtěl ještě zmínit, je nedostatek freewarových softwarů pro jednoduché vytváření formulářů. Podle mého názoru je velmi dobrým softwarem „Adobe LiveCycle Designer“, který umožňuje opravdu snadné vytváření PDF formulářových šablon, přidávání validací a kalkulací bez znalosti programování. Software je součásti balíku Adobe Acrobat 9 Pro, a volně dostupná je bohužel jen trial verze na 30 dnů. Cena tohoto softwaru není příliš příznivá. Pohybuje se okolo 600 euro.

Pokud shrneme zmiňovaná pro a proti, je zřejmé, že PDF formuláře mají dostatečné výhody oproti HTML formulářům.

2.6 Nástroje pro práci s formuláři

Nejznámějším nástrojem pro práci s PDF formuláři je již zmíněný Adobe Acrobat a Adobe LiveCycle Designer, které se dají rozšiřovat o množství různých pluginů.

Pro jednoduché prohlížení a vyplňování formulářů zcela postačí freewarový Acrobat Reader.

Z nekomerčních aplikací, pomocí kterých se dají vytvářet PDF formuláře bych zmínil řádkově orientovaný TeX, který musí být doplněn o příslušné převodníky (např. dvipdfm) nebo jeho variantu pdfTex.

Vytvářet PDF formuláře pomocí programovacího prostředí lze také prostřednictvím knihoven. Může se jednat o komerční knihovnu „PDFLib“, viz použitá literatura [4], nebo nekomerční javovskou knihovnu „iText“, viz [5], kterou používám v diplomové práci.

Nástroje pro zpracování dat jsou například FDF Toolkin, který je překvapivě od společnosti Adobe. Umožňuje vytvářet server jak pro Unix, tak pro Windows NT v jazycích

Java, Perl C/C++ a dalších. Jsou-li data odesíláná v HTML podobě, lze je klasicky zpracovávat pomocí příslušných skriptů v Javě, PHP, Ruby atd. Pro zpětnou odezvu je ale zapotřebí modifikace do FDF formátu.

Přehled dalších nástrojů je k naleznutí na webu PlanetPDF[6] či PDFzone[7].

3 CouchDB

Apache CouchDB je dokumentově orintovaná databáze, která neskladuje data v tabulkách se stejnou strukturou polí pro každý záznam. Místo toho je každý záznam uložen jako dokument, který má určité vlastnosti. Do těchto dokumentů můžete přidávat libovolný počet polí/hodnot s jakoukoliv délkou. Může být dotazováná a indexováná v takzvané MapReduce formě s použitím JavaScriptu.

Je založena na RESTful JSON API, ke kterému můžete mít přístup z jakéhokoliv prostředí podporujícího HTTP požadavky. Jádro je napsáno v jazyce Erlang, který vyvinula firma Ericson a roku 1998 jej uvolnila jako open source. Erlang je robustní funkcionální jazyk, vhodný pro tvorbu souběžných distribuovaných systémů.

PDF formůlaře jsou v základě dokumenty, proto jsem se rozhodl využít v projektu vlastností CouchDB.

V následujicích odstavcích podrobněji vysvětlím několik základních rysů CouchDB. Více informací naleznete na domovských stránkách tohoto projektu [8].

3.1 Schéma Databáze

Jedním z problémů, kterým se zabývájí tradiční relační databáze jako SQL je, že při jakékoliv změně či přidání attributu pro objekty, které chceme uchovávat v databázi, musíme změnit i schéma dané databáze. Někdy si vystačíme s příkazem ALTER TABLE, jindy je zapotřebí naprogramovat složitější skript, operace atd. CouchDB oproti tomu schéma nemá. Je takzvaně „schema-free“, čímž umožnuje ukládat volné struktury. Podle mého názoru tato vlastnost není vždy výhodou. Například pokud máme data vysoce strukturovaná, mezi ruznými třídami existují složité relace a jejich podoba se příliž nemění - v tomto případě mohou být relační databáze lepší volbou. Pro dokumentově orientovanou databázi je však bezschématový přístup na pravém místě. V následujícím textu vysvětlím proč.

Dokumenty v reálném světě:

- platby
- daňové formuláře
- dopisy
- výpisy

Je všeobecně známo, že tyto dokumenty patří k nejvíce používaným dokumentům v reálném světě. Platby, výpisy a daňová přiznání používá každá firma. V čem ale tkví problém? Přestože můžeme říci, že se jedná o dokumenty stejného typu, po zhlédnutí několika plateb od různých firem zpozorujeme určitou rozdílnost ve struktuře dokumentů. Z této skutečnosti lze vyvodit základní informaci, která nám mnohdy uniká.

„stejný typ != stejná struktura“

„Out Of Date“
Dalším rysem dokumentů je to, že se často mění podle potřeb firem, zákonů atd., což může v relačních databázích přivodit složitý proces pro změnu schématu databáze. V dokumentově orientované databázi postačí změnit pouze strukturu reprezentativních tříd k těmto dokumentům a zaintegrovat logiku pro zpracování starých atributů do nových. Tyto změny provádíme pouze na straně aplikace.

Přirozené chování dat
Nejdůležitějším aspektem pro pochopení rozdílu mezi dokumentově orientovanými a relačními databázemi je skutečnost, že v dokumentově orientovaných DB jde pouze o dokument. Ještliže dokument obsahuje data, která jsme byli zvyklí v relačních databázích skladovat v jiných tabulkách a odkazovat na ně pomocí cizích klíčů (např. kategorie, číselníky měst . . .), do dokumentu je prostě vložíme. K dokumentům v těchto databázích přistupujeme jako k dokumentům v reálném světě. Představte si, že byste četli dokument, kde bude „Město: 23“. To ovšem neznamená, že v těchto databazích nemůžeme využívat relací. Jestliže je relací zapotřebí, měli bychom si dobře promyslet, jaký typ databáze zvolíme.

3.2 JSON

Než budu pokračovat v popisu dalších rysů CouchDB, chtěl bych zmínit základní datový formát, se kterým CouchDB pracuje *JSON*.

JavaScript Object Notation (JavaScriptový objektový zápis). JSON je datový formát nezávislý na platformě, který je vyvinutý pro přenos dat. Je údajným nástupcem XML formátu. Mnozí vývojáři při srovnávání JSONu a XML vidí velkou výhodu v tom, že JSON není tak „ukecaný“. Data mohou být organizováná v polích nebo agregováná v tzv. objektech (podobné hash – asociativním polím). Složitost této hiearchie je neomezená. JSON formát se hojně používá v AJAX dotazech, dokáže jej zpracovat libovolný programovací či skriptovací jazyk.

JSON umí pojmout pole hodnot, objekty, speciální prázdnou hodnotu *null* a hodnoty, kterými mohou být celá i desetinná čísla, řetězce a booleovské hodnoty *true*, *false*.

```
{
   "attachment_describe": "describe_test.pdf",
   "name": "test",
   " original_fields ": [
      {
         "name": "form1[0].#subform[0].first_name[0]",
         " llx ": 199,
         " lly ": 679,
         "default_values": null,
```

```
                "urx": 298,
                "type": " textfield ",
                "page": 1,
                "ury": 699
            },
        ],
        "created_at": "2010/03/07 11:27:18 +0000",
        "forked": true,
        "updated_at": "2010/03/07 11:27:18 +0000",
        "attachment": "test .pdf",
        "done": true,
        "user": "nevralaz@gmail.com",
        }
    }
```

Výpis 1: Ukázka JSON formátu

3.3 RESTful HTTP API

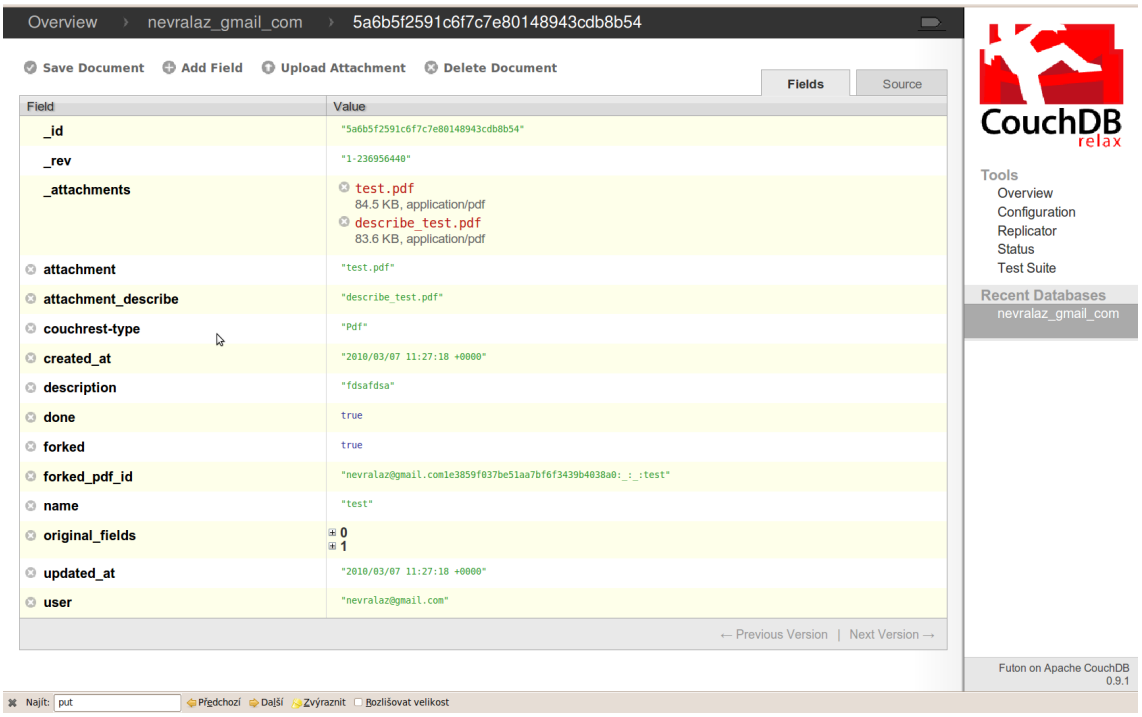
Vpřípadě, že jste někdy měli problém s instalací ovladačů pro MySQL nebo PostgreSQL na různých webových platformách, u CouchDB se tomu tak určitě vyhnete. CouchDB komunikuje vyloženě přes HTTP. Chcete-li vytvořit novou databázi nebo dokument, stačí poslat HTTP PUT dotaz se zadanými daty. Chcete-li získat dokument z databáze, stačí poslat HTTP GET dotaz. Pomocí HTTP DELETE můžete dokumenty mazat. Jak je vidět API je poměrně jednoduchá a pokud neexistuje klientská knihovna pro práci s CouchDB ve vámi zvoleném jazyce, její napsání zabere jen několik minut. Velmi zajimavé je, že ke spolupráci s databází stačí pouze HTML stránka s AJAX dotazi na databázi. Můžeme tak sestavit velmi jednoduchou aplikaci pro tzv. CRUD (create, read, update, delete) nějakého modelu. Tímto odpadá nutná spolupráce s programovacím či skriptovacím jazykem na serverové straně.

Seznam CRUD pro API

Create:	HTTP PUT	/db/docid
Read:	HTTP GET	/db/docid
Update:	HTTP POST	/db/docid
Delete:	HTTP DELETE	/db/docid

3.4 Dokumenty

Dokumenty jsou v databázi uloženy ve zmiňovaném JSON formátu. CouchDB poskytuje uživatelské rozhraní jménem Futon pro procházení, prohlížení, úpravu a mazání dokumentů a databazí. Na obrázku č.1 vidíme, jak vypadá dokument ve Futonu. Dokument má tři názvy atributů, rezervovány pro CouchDB, jsou to: „_id“, „_rev“, „_attachments“. Jejich popis vysvětlím níže.



Obrázek 1: Futon - dokument

3.4.1 Unikátní ID, Revision number

Každý dokument v CouchDB má dva povinné atributy: _id a _rev. Jak jsme již zvyklí, id dokumentu (_id) musí být unikátní. Pokud si id neurčíme při vkládání nového dukumentu, CouchDB nám jej vygeneruje samo jako tzv. UUID. Jde o unikátní klíč, který zaručí, že nebude vygenerován v jiné databázi. Automatické generování UUID je tedy velmi vhodné pro distribuované databáze.

Atribut _rev označuje revizi. Používáme jej hned k několika účelům. Prvním je řešení současného přístupu k datům. Jestliže chceme dokument v CouchDB změnit, posíláme jej i s atributem _rev. Dokument se po změnách uloží jen tehdy, pokud číslo revize měněného dokumentu odpovídá aktuálně platnému číslu revize dokumentu v databázi. Při každé změně dokumentu se změní i číslo revize. Druhým použitím _rev atributu je listování v historii změn v dokumentu. CouchDB totiž umožnuje procházet historii změn v dokumentu podle tohoto čísla revize.

3.4.2 Přílohy

Jako většina dokumentů, mohou mít i dokumenty v CouchDB přílohy. Posíláme je v těle požadavku ve formátu JSON spolu s ostatními daty. Atribut pro přílohy se jmenuje _attachments. Ve výpisu kódu 2 vidíme, že _attachments obsahuje přílohy jako další JSON

objekty, kde klíč je jméno přílohy a ta obsahuje atributy content_type (typ přílohy) a data (příloha zakódována v BASE64).

```
{
    "_attachments" : {
        "front_cover.jpg" : {
            "content_type" : "image/jpeg",
            "data" : "/9j/4AAQSkZJRgABAQEAnQCdA..."
        },
        "back_cover.jpg" : {
            "content_type" : "image/jpeg",
            "data" : "/AQS54SD66778A..."
        }
    },
    " title " : "Dokument",
    "genre" : "action"
}
```

Výpis 2: Přílohy

3.5 MapReduce

Pro dotazování používá CouchDB JavaScript a tzv. pohledy. Pohledy jsou definovány dvěma funkcemi „map“ a „reduce“. MapReduce je programovací model pro zpracování a generování velkých množin dat, vyvinutý společností Google. Funkce map je povinná, slouží k filtrování hledaných dat a volá se pro každý dokument. Jestliže dokument odpovídá požadovaným parametrům, podle kterých map vyhledává, vygeneruje dvojici klíč / hodnota. Funkce reduce není povinná a slouží k agregování. Pokud je funkce reduce naimplementována v pohledu, výsledky z funkce map se vyberou, vyrobí se z nich množina dvojic klíč / seznam všech hodnot mapovaných pro klíč. Pro každou z těchto dvojic se zavolá redukční funkce. Můžeme tak například vytvořit redukční funkci, která spočítá sumu všech zakoupených knih od určitého spisovatele.

Ve výpisu kódu 3 vidíme pohled s funkcemi map i reduce. Funkce map vygeneruje pro každou knihu v každém žánru jednu jedničku. Funkce emit generuje dvojici klíč / hodnota. Funkce map předá funkci reduce seznam klíčů s hodnotami přiřazenými k těmto klíčům. Ve funkci reduce sečteme hodnoty, které se přiřadili ke klíčům (v našem případě jedničky), tak dostaneme počet knih přidělených k jednotlivým žánrům.

```
// map
function(doc) {
  if (doc.type == "book" && doc.genres) {
    for (var genre in doc.genres) {
      emit(doc.genres[genre], 1);
    }
  }
}

// reduce
function(keys, values, rereduce) {
```

```
    return sum(values);
}
```

Výpis 3: Ukázka MapReduce pohledu

Pohledy se ukládají do databáze a dají se volat přes URL. Jako parametry můžete navíc použít:

- key=keyvalue
- startkey=keyvalue
- startkey_docid=docid
- endkey=keyvalue
- endkey_docid=docid
- limit=maximální počet řádků pro vrácení
- descending=true
- skip=počet řádků pro přeskočení

Například parametr „?startkey=the“ přidaný do URL pohledu bude navíc filtrovat jen ty výsledky z funkce map, u nichž klíč začíná na řetězec „the“.

4 Resque

Resque je Redis backendová knihovna pro vytváření úkolů zpracovávaných na pozadí. Redis je open source key-value databáze. Pracuje podobně jako memcache, kde hodnoty nemusí být pouze řetězce, ale také seznamy, sady nebo setřízené sady. Všechny tyto datové typy mohou manipulovat se základními operacemi jako push/pop (přidat/odebrat) prvek. Proto je Redis velmi dobrý nástroj pro systém front s úkoly.

Úkoly se vkládají do front a zpracovávají se pomocí tzv. workerů[3]. U každého workera můžeme určit, kterou frontu bude sledovat a zpracovávat. Resque také nabízí webové rozhraní pro monitorování front, workerů, chyb atd. Na obrázku 2 můžeme vidět webové rozhraní, na němž je zobrazený seznam front a úkolů čekajících na zpracování workery.

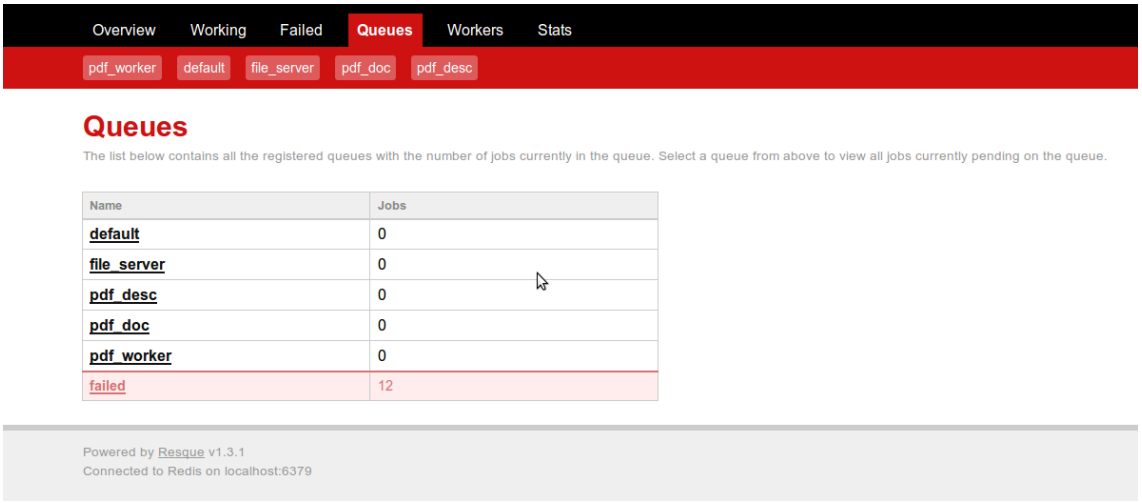
Práce s PDF soubory zaberou více systémového času než jiné akce v portálu. Při větším množství uživatelů, kteří budou požadovat ve stejný čas práci s PDF souborem, může nastat zahlcení serveru požadavky, které nebude stíhat zpracovávat. V takovéto situaci hrozí možnost, že některým požadavkům vyprší čas pro zpracování. Proto jsem se rozhodl využít knihovnu Resque a její možnost vkládat úkoly do front k asynchronnímu zpracování pro workery.

Později se ukázala další výhoda využití této knihovny v mé práci. Jedinou volně dostupnou knihovnou pro zpracování PDF formulářů je „iText“, která je napsána v Javě. Původně jsem aplikaci vyvíjel pomocí tzv. ruby-java-bridge, který mi umožňoval v Ruby třídách načítat java knihovny a pracovat s java kódem. Tato cesta se však ukázala jako slepá. Ruby-java-bridge měl vysoké nároky na používanou paměť. Kromě toho z doposud neobjasněného důvodu vytvářel na serveru nekonečnou smyčku, která se vyskytla, jakmile se spouštěl kód používaný ruby-java-bridge.

Proto jsem se uchýlil k řešení, ve kterém jsem rozdělil aplikaci na dvě části. První částí je webová aplikace, která umožňuje práci se šablonami, dokumenty, uživateli adt. Je napsána v Ruby on Rails. Druhá část jsou workeři, kteří zpracovávají PDF soubory. Ti vytvářejí popisné PDF k originálnímu PDF formuláři, a PDF formuláře s vyplněnými daty od uživatele (viz příručka uživatele). K těmto úkolům je zapotřebí iText (java) knihovny. Workeři jsou napsáni pro JRuby, což je implementace programovacího jazyka Ruby v Javě. Komunikační linkou mezi webovou aplikací a samotnými workery je Resque knihovna spolupracující s Redis serverem, na němž se ukládají úkoly pro workery. Zde mám vytvořeny fronty pro jednotlivé úkoly.

Na obrázku 3 jsem jednoduchou formou znázornil, jak výše zmiňované workflow funguje. Je na něm znázorněno, jak webová aplikace(2) při požadavku vygenerování PDF vloží tento úkol do fronty v Redis(1) s informacemi o PDF v databázi(4). Workeři periodicky kontrolují fronty s úkoly v Redis. Pokud nějaký vyzvednou, zpracují požadovaná data z databáze a vygenerují PDF, které následně uloží zpět do databáze.

[3] worker - pracovník



Obrázek 2: Resque - webové rozhraní

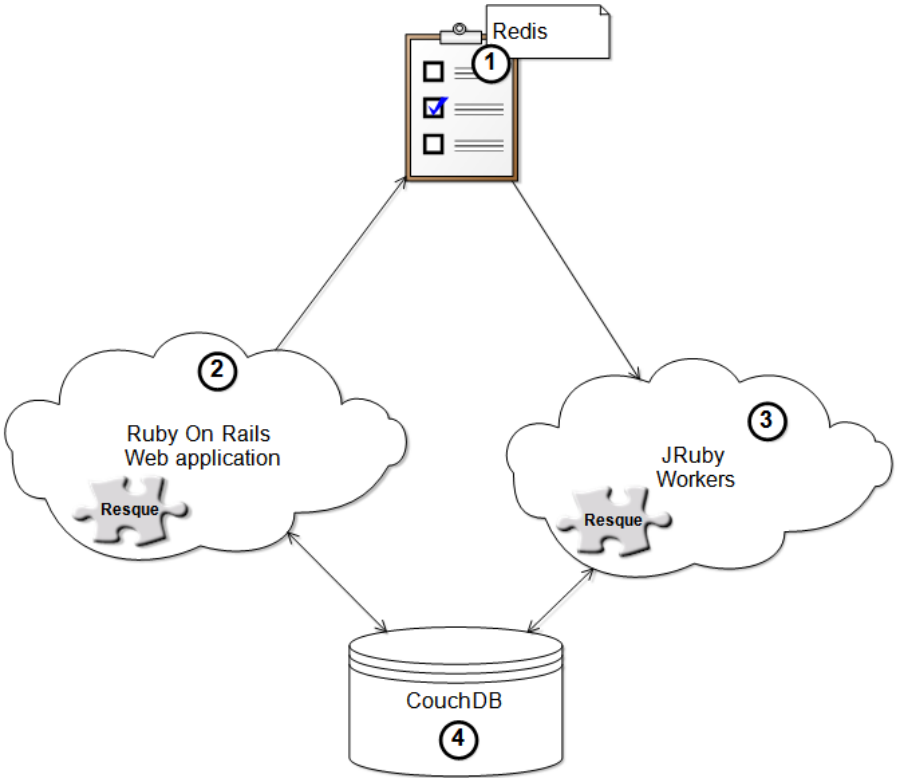
Ve výpisu 4 uvádím obsah konfiguračního souboru pro resque workery, který nalezneme v „config/resque.yml“. Položka INTERVAL nastavuje interval pro workery, v jakém mají kontrolovat fronty s úkoly pro zpracování. Položka QUEUES určuje fronty s úkoly, které mají workeři zpracovávat. Pomocí položky WORKERS nastavíme počet workerů, kteří fronty zpracovávají. VERBOSE a VVERBOSE (very verbose) jsou nastavení pro logování workerů na standardní STDOUT.

```
INTERVAL: 5
VERBOSE: false
VVERBOSE: true
QUEUES:
- pdf_desc
- pdf_doc
WORKERS: 5
```

Výpis 4: Nastavení Resque - resque.yml

Ruční spuštění workerů na pracovní stanici můžeme provést pomocí rake tasku „rake resque:start_jworkers“. Na serveru jsou tito workeři spuštěni na pozadí.

Více informací o knihovně Resque najdeme na http://github.com/defunkt/resque



Obrázek 3: Workflow

5 Návrh a analýza pro PDF portál

Portál pro PDF formuláře by měl poskytovat základní možnosti pro práci s formuláři. To znamená nahrávání PDF formulářů, vytváření vlastních šablon pro PDF formuláře, editování těchto šablon. Dále vytváření, prohlížení, editování a mazání dokumentů k formulářům. Všechny výše zmíněné úpravy může uživatel provádět privátně na svém soukromém účtu, v takzvaných vlastních repozitářích. Portál by měl poskytovat možnost nahrávání PDF formulářů a šablon do veřejného repozitáře. Z něj si je ostatní uživatelé budou moci nahrát na vlastní účet. Nahrávání mezi účty bude probíhat pomocí takzvaného „forkování“[4]. Jestliže bude mít uživatel vypracovanou šablonu formuláře, (např. pro vyplnění daní na rok 2010) v určité fázi, a bude ji chtít poskytnout ostatním uživatelům ke stažení, forkne (zkopíruje) jí do věřejného repozitáře, odkud si ji budou moci ostatní uživatelé forknout do vlastního repozitáře. Portál bude obsahovat „tagy“, které budou přidělovány k formulářům a šablonám. Tagy budou sloužit k bližšímu určení kategorie daného formuláře či šablony, např. „smlouvy“, „daně“, „plné moci“... Šablony pro PDF formuláře by měly mít možnost provádět jednoduché kalkulace, které umožní dopočítat matematické závislosti mezi jednotlivými políčky pro vyplnění. Šablony by neměly postrádat základní druhy validace, např. nutnost vyplnění políčka nebo podmínku, že vyplňované políčko musí obsahovat číslo.

Administrátor má navíc možnost spravovat veřejné repozitáře a tagy. Dále může spravovat uživatelské účty, zrušit je, nebo přidělovat práva jiným uživatelům.

Vice informací viz uživatelská a administrátorská příručka na přiloženém CD.

5.1 Návrh

Moje řešení je založeno na dvou typech databází, na relační MySQL a dokumentově orientované CouchDB. Původně jsem zamýšlel v CouchDB uchovávat jen informace spjaté s PDF formuláři, šablonami a dokumenty vyplněnými v šablonách díky jejich spjatosti s dokumenty. Informace o uživatelích a kategoriích jsem zamýšlel skladovat v MySQL. Kategorie pro formuláře měly být stromové struktury - pro vztahy mezi kategoriemi by tedy byla vhodná relační databáze. V průběhu vývoje jsem se ale rozhodl pro řešení pomocí tagů (značek), které se mi zdálo jednodušší, přehlednější a lépe spravovatelné. Tagy skladuji v CouchDB, proto v relační databázi zbývá pouze tabulka uživatelů, rolí a vazební tabulka mezi uživali a rolemi.

Tabulka **Users** definuje fyzickou osobu, její email a přihlašovací údaje. Dokumentově orientované databáze neobsahují tabulky ale dokumenty. Výhoda těchto dokumentů je, že jejich struktura nemusí být pevně stanovena a může se měnit. Přesto jsem pro jednotlivé typy dokumentů nadefinoval základní strukturu, která popisuje typ těchto dokumentů. Dokument **Tag** obsahuje jméno tagu, jeho popis, seznam pdf a šablon, které jsou označeny tímto tagem, dále pak počet pdf a šablon. Dokument **Pdf** obsahuje jméno,

[4] fork - rozvětvení (v našem případě zkopírování)

popis pdf, nahraný PDF formulář, vytvořený „pdf describe“, což je vygenerovaný PDF dokument s popisky jmen jednotlivých polí pro vyplnění k pozdějšímu vytváření šablon. Dokument obsahuje seznam polí pro vyplnění formuláře a **PdfTemplate** reprezentuje šablonu, kterou vytváříme pro Pdf. Obsahuje jméno, popis, id pdf z něhož se šablona vztahuje, seznam polí, které jsme se rozhodli v naší šabloně z původního formuláře použít. PdfTemplate dále obsahuje skupiny, pomocích nichž utváříme vzhled šablony. Skupiny obsahují jednotlivá pole a mají jméno, což umožňuje šablonu rozdělovat na jednotlivé tématické sekce jako například: osobní informace, bankovní informace atd. Kromě toho šablona obsahuje seznam nadefinovaných matematických kalkulací, sloužících pro výpočty závislostí mezi jednotlivými poli. Pole v šablonách mají také svoji strukturu:

- jméno a popis, pod jakým se v šabloně zobrazují
- typ pole: textfield, checkbox, radiobutton ...
- jméno originálního pole v PDF formuláři
- defaultní hodnotu, která se bude automaticky vyplňovat při vytváření nového dokumentu ze šablony
- seznam validátorů určujících chování pole v šabloně (např. jestli má být pole vyplněno, či může obsahovat pouze číslo)

PublicPdf a **PublicPdfTemplate** jsou dokumenty určené pro veřejný repozitář obohacené o tagy a logiku, pomocí které můžeme **Pdf** a **PdfTemplate** dokumenty forkovat do veřejných repozitářů.

Implementaci jsem se rozhodl řešit v jazyce Ruby a frameworku Ruby on Rails, který je založen na návrhovém vzoru Model View Controler (MVC). Tento jazyk mi umožňuje jednoduchou práci s modely(třídami), definování jejich callback metod, jednoduchou práci s vazbami mezi modely, validacemi a dobrou integraci testů viz kapitola 8.

5.2 Datová analýza

Před kompletací tohoto úseku bylo velmi důležité utvořit si obecný přehled o požadovaných informacích pro tvorbu šablon a dokumentů a sestavit ucelený přehled informací, které budou v systému evidovány.

5.2.1 Lineární zápis

(**primární klíč**, *cizí klíč*)

Users (**id**, email, password_hash, couchdb_name, lock_version, created_at, updated_at)

User_roles (**id**, *user_id*, *role_id*, created_at, updated_at)

Roles (**id**, name, description, created_at, updated_at)

Pdf (**_id**, _rev, _attachments, attachment, attachment_describe, couchrest-type, name, description, done, forked, *forked_pdf_id*, original_fields, *user*, created_at, updated_at)

PdfTemplate (**_id**, _rev, calculations, couchrest-type, name, description, fields, forked, *forked_template_id*, groups, *pdf_id*, *user*, created_at, updated_at)

Document (**_id**, _rev, _attachments, couchrest-type, name, done, values, *pdf_template_id*, created_at, updated_at)

PublicPdf (**_id**, _rev, _attachments, attachment, attachment_describe, couchrest-type, name, description, done, forked, forked_no, *forked_pdf_id*, original_fields, tags, *user*, created_at, updated_at)

PublicPdfTemplate (**_id**, _rev, calculations, couchrest-type, name, description, fields, forked, *forked_template_id*, groups, *pdf_id*, *user*, tags, created_at, updated_at)

Tag (**_id**, _rev, couchrest-type, name, description, pdfs, pdfs_no, templates, templates_no, created_at, updated_at)

Field (name, description, type, default_value, default_values, original_name, page, validators)

Group (name, fields)

Calculatios (name, calculation, prePopulate)

Validator (klass, checked, message)

5.2.2 Schéma databází

Jelikož se PDF portál skládá ze dvou typů databází, zvolil jsem individuální nákres těchto databází namísto klasického E-R diagramu. Na obrázku 8 vidíme v levé části klasickou relační databázi s tabulkami a na pravé straně databáze CouchDB s dokumenty. Databáze jsou rozděleny na jednu veřejnou(1), kde se vyskytují tagy, šablony a pdf volně dostupná uživateli ke stažení, a soukromé databáze(2), které jsou vytvořeny jednotlivě ke každému uživatelskému účtu. V soukromých databázích mají uživatelé uloženy svoje pdf, šablony a dokumenty.

5.2.3 Tabulky a popisy atributů

U jednotlivých tabulek popisuji některé důležité atributy pro lepší pochopení úlohy atributů v systému. Zbytek atributů je popsán v příloze tabulky. Kontrola povinných atributů je řešena na aplikační úrovni na straně Ruby.

5.2.3.1 Tabulka Users Tabulka Users obsahuje informace o přihlášeném uživateli. Povinné údaje jsou: email, password_hash, couchdb_name. Více atributů (např. adresu, telefon, ...) jsem neimplementoval, neboť k základnímu požadavku funkčnosti PDF portálu nejsou zatím potřebné.

email - je email uživatele, pod kterým se bude přihlašovat do systému. Email musí být v databázi jedinečný. Podle emailu se uživateli vygeneruje privátní databáze v CouchDB pro skladování vlastních pdf, šablon a dokumentů.

couchdb_name - je jméno vygenerované dokumentové databáze pro uživatele.

Tabulka s popisem atributů viz tabulka 2

5.2.3.2 Tabulka Roles a User_roles Tabulka roles je číselník, obsahující seznam rolí, pod kterými může přihlášený uživatel vystupovat v systému s nastavenými právy pro tyto role. Tabulka User_roles je vazební tabulkou mezi uživatelem a rolemi. V systému může mít uživatel nastavenu pouze jednu roli, proto se může toto řešení s vazební tabulkou zdát zbytečné. Při implementaci jsem se ale přiklonil k tomuto způsobu pro případ, že by v budoucnu z nějakého důvodu mohla být požadována možnost kombinace rolí pro uživatele.

V systému jsou dva typy rolí registrovaných uživatelů:

- admin - uživatel s právy administrátora, který má možnost spravovat veřejný repozitář, tagy a uživatele
- registered - registrovaný uživatel s právy forkovat z/do veřejného repozitáře a spravovat vlastní repozitář (pdf, šablony, dokumenty)

Tabulky s popisem atributů viz tabulky 3 a 4

5.2.4 Dokumenty a popisy atributů

U CouchDB dokumentů jsem pomocí modelů nastavil strukturu jednotlivých dokumentů. Navíc jsem vytvořil modely, které nejsou samotnými dokumenty, ale jejich součástí. Tyto modely mají naimplementovánu vlastní logiku i chování a do dokumentu se ukládají v podobě hash pole jako nastavený atribut. Po vytažení dokumentu z databáze se hash atributy opět změní v model, který je u atributu definován, a získají zpět svoje specifické chování i logiku. Toto chování nastavujeme v modelu dokumentu pomocí definování atributu s parametrem „:cast_as=> 'jmeno_modelu'“ nebo „:cast_as => ['jmeno_modelu']“, což určí, že atribut bude objekt 'jmeno_modelu' nebo pole objektů 'jmeno_modelu'. Zmínku o řešení atributů v datové analýze jsem uvedl z důvodu lepšího pochopení následujících informací. Podobně jako u popisů tabulek popíšu pouze důležitější atributy. Tabulky s celými strukturami dokumentů jsou umístěny v příloze Tabulky.

5.2.4.1 Dokument Pdf Dokument Pdf obsahuje základní informace o nahraných PDF formulářích. Povinné atributy jsou: name, _attachments, attachment, attachment_describe, user.

name - pojmenování PDF formuláře, který chcete nahrát do systému

attachment - jméno nahraného PDF formuláře

attachment_describe - jméno vygenerovaného PDF formuláře s popisky k jednotlivým polím.

_attachments - hash pole obsahující nahrané PDF soubory (PDF formulář s PDF describe formulářem). Nahrané soubory musí být typu 'application/pdf'. Jelikož jsou přílohy PDF formulářů uloženy v hash poli, k ukládání jmen využíváme atributy attachment a atttachment_describe , pod kterými je v tomto hashi najdeme.

user - tento atribut určuje uživatele v systému, který pdf nahrál do systému. Ne vždy to je uživatel, který má dokument ve vlastním repozitáři, neboť pdf můžeme forkovat z veřejného repozitáře.

original_fields - hash pole originálních polí v PDF formuláři. Každé pole obsahuje tyto informace:

- name - originální jméno pole
- default_values - výchozí hodnoty pole ve formuláři. Používáno pro pole radiobuttonů, kdy se několik radiobuttonů v poli chová jako jeden, pouze hodnoty se liší
- type - typ pole
- page - číslo stránky na které se pole vyskytuje ve formuláři
- llx - x souřadnice levého dolního rohu pole
- lly - y souřadnice levého dolního rohu pole
- urx - x souřadnice pravého horního rohu pole
- ury - y souřadnice pravého horního rohu pole

forked_pdf_id - pdf dokument má přidělen speciální id pro forkování do veřejného repozitáře, pomocí kterého bude unikátní ve všech repozitářích. Používá se pro zabránění nadbytečného forkování stejných PDF dokumentů do repozitářů.

Dokument s popisem atributů viz tabulka 5

5.2.4.2 Dokument PublicPdf Co se týče datové struktury, je Dokument PublicPdf téměř stejný jako dokument Pdf. Jsou u něj navíc přidány některé atributy, potřebné pro skladování dokumentů ve veřejném repozitáři.

Dokument s popisem atributů viz tabulka 6

5.2.4.3 Dokument PdfTemplate Dokumnt PdfTemplate obsahuje informace o námi vytvořené šabloně k pdf. Povinné atributy jsou: name, pdf_id, user.

user - určuje uživatele, který šablonu vytvořil

forked_template_id - pro forkování šablony do veřejného repozitáře je šabloně vytvořeno speciální forkovací id, aby se zabránilo duplicitě šablon v repozitářích (stejně jako forked_pdf_id u dokumentu Pdf). Pokud však uživatel provede změny v šabloně a poté chce šablonu opět forkovat do systému, musí změnit i jméno šablony, aby bylo zřejmé, že se v šabloně provedly změny. Poté ji může znovu forkovat do veřejného repozitáře.

fields - je pole složené z objektů(modelů) typu Field viz níže

groups - pole složené z objektů typu Group viz níže. V šabloně si uživatel může vytvářet skupiny, do nichž začleňuje jednotlivá pole. Tímto způsobem si uživatel vytvoří vlastní přehledný vzhled šablony skládající se z logických skupin definovaných uživatelem (skupina osobní informace, skupina bankovní účty . . .)

calculations - pole složené z objektů typu Calculation viz níže

Dokument s popisem atributů viz tabulka 7

5.2.4.4 Dokument PublicPdfTemplate Dokument PublicPdfTemplate má podobnou strukturu jako dokument PdfTemplate. Jsou u něj přidány informace potřebné pro šablony ve veřejném repozitáři.

Dokument s popisem atributů viz tabulka 8

5.2.4.5 Dokument Document Dokument Document obsahuje informace o vyplněné šabloně. Má v sobě uložený vygenerovaný PDF formulář s vyplněnými daty od uživatele a vyplněná data sama o sobě pro případnou editaci. Povinné atributy jsou: name, pdf_template_id.

values - hash pole obsahující originální názvy polí v PDF formuláři s hodnotami pro vyplnění.

Dokument s popisem atributů viz tabulka 9

5.2.4.6 Dokument Tag Dokument Tag zastupuje tzv. značku pomocí které můžeme pdf a šablony značkovat a tím určovat, do kterých kategorií patří. Povinným atributem je „name“.

pdfs - pole pdf id, které byly přiděleny k tagu

pdfs_no - počet pdf přidělených k tagu

templates - pole id šablon, které byly přiděleny k tagu

templates_no - počet šablon přidělených k tagu

Dokument s popisem atributů viz tabulka 10

5.2.4.7 Model Field Model je částí dokumentů PdfTemplate a PublicPdfTemplate. Reprezentuje pole s vlastnostmi nadefinovanými uživatelem pro šablonu. Pole odpovídá originálnímu poli v PDF formuláři. Uživatel mu může přidělit vlastní jméno, defaultní hodnotu, typ (textfield, checkbox ...), přidělit validace (musí být vyplněno, musí být číslo). Povinnými atributu jsou: name, original_name.

name - jméno pole pro zobrazení v šabloně - je v šabloně unikátní.

original_name - jméno originálního pole v PDF formuláři

default_value - defaultní hodnota

default_values - v PDF formuláři může nastat situace, že pole je skupina radiobuttonů. V tomto případě to budou hodnoty radio buttonů, které může uživatel spravovat (přejmenovávat)

validators - pole validátorů vztahující se na pole (viz níže)

Model s popisem atributů viz tabulka 11

5.2.4.8 Model Group Tento model je součástí dokumentů PdfTemplate a PublicPdfTemplate. Model reprezentuje pojmenovanou skupinu polí v šabloně. Povinným atributem je „name“.

name - jméno skupiny pro zobrazení v šabloně - je v šabloně unikátní.

fields - pole se jmény polí vytvořenými v šabloně.

Model s popisem atributů viz tabulka 13

5.2.4.9 Model Calculation Model Calculation je taktéž součástí dokumentů PdfTemplate a PublicPdfTemplate. Obsahuje potřebné informace pro výpočet jednoduchých kalkulací mezi poli. Povinnými atributy jsou: name, calculation.

calculation - matematický vzorec pro výpočet kalkulace

prePopulate - pole terminálů a nonterminálů (blíže v kapitole 6.1) použitých ve vzorci pro našeptávač pomocí kterého můžeme editovat kalkulaci

Model s popisem atributů viz tabulka 12

5.2.4.10 Model Validator Validator je součástí modelu Field. Obsahuje potřebné informace pro validaci pole. Povinný atribut je „klass“.

klass - určuje typ validátoru. V systému existují dva typy validátorů PresenceValidator a NumberValidator

message - zpráva, která se zobrazí, pokud validace vrátí false

Model s popisem atributů viz tabulka 14

5.3 Funkční specifikace

Dalším krokem při vývoji PDF portálu je funkční specifikace. Řeším ji za pomoci diagramů případů užití. Diagramy určují případy užití portálu a aktéry, kteří vstupují do interakce s portálem a stojí vně portálu.

5.3.1 Seznam primárních aktérů

V tomto kroku jsem určil seznam aktérů, který jsem odvodil z požadavků na portál (viz výše).

Seznam aktérů
admin
registrovaný uživatel
neregistrovaný uživatel

Tabulka 1: Seznam aktérů

5.3.2 Případy užití

Z navržených požadavků pro portál (viz začátek této kapitoly) jsem sestavil základní případy užití. Diagramy jsou k nahlédnutí v příloze A. Pro zpřehlednění diagramů jsem sloučil některé případy užití do jednoho, tzn. například ve správě veřejného pdf, toto pdf editujeme a následně smažeme.

5.3.2.1 Registrace nového uživatele viz obrázek 9

Aktéři Neregistrovaný uživatel

Úroveň Uživatelský cíl

Rozsah Pdf portál

Vstupní podmínky - - -

Záměr Registrace uživatele do systému PDF portál

Hlavní scénař

1. uživatel zvolí možnost registrace nového uživatele
2. systém zobrazí registrační formulář
3. uživatel vyplní požadované informace
4. systém zobrazí uživateli přihlašovací stránku s informací o registraci

Rozšíření

3a systém validuje email a heslo
   1. jetliže email v systému již existuje, systém přejde na krok 2
   2. jestliže nejsou hesla validní, systém přejde na krok 2

Formulář pro registraci

- email - textfield
- password - textfield
- password confirmation - textfield
- register - button

5.3.2.2 Přihlášení Viz obrázek 10

Aktéři Neregistrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky - - -

Záměr Přihlášení uživatele do systému PDF portál

Hlavní scénař

1. uživatel navštíví stránku pro přihlášení
2. systém zobrazí přihlašovací formulář
3. uživatel vyplní přihlašovací údaje a odešle
4. systém zobrazí úvodní obrazovku pro přihlášeného uživatele

Rozšíření

3a systém validuje email a heslo
   1. pokud v systému neexistuje účet se zadaným emailem a heslem, přejde systém na krok 2

Formulář pro přihlášení

- email - textfield
- password - textfield
- login - button

5.3.2.3 Vytvoření tagu viz obrázek 9

Aktéři Admin

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému a má práva administrátora

Záměr Administrátor přidá do systému nový tag

Hlavní scénař

1. uživatel zvolí možnost vytvoření nového tagu
2. systém zobrazí formulář pro vytvoření tagu
3. uživatel zadá jméno a popis tagu a formulář odešle
4. systém zobrazí seznam tagů obsahující vytvořený tag

Rozšíření

a pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

3a systém validuje zda jméno tagu již neexistuje v systému

1. pokud v systému jméno tagu existuje, systém přejde ke kroku 2

Formulář pro vytvoření tagu

- name - textfield
- description - textarea
- create - button

5.3.2.4 Správa uživatele viz obrázek 9

Aktéři Admin

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému a má práva administrátora

Záměr Administrátor změní uživateli práva v systému

Hlavní scénař

1. uživatel zvolí možnost editovat uživatele ze seznamu uživatelů
2. systém zobrazí formulář s informacemi o uživateli
3. uživatel uživatel zvolí ve formuláři, zda je uživatel administrátor a odešle formulář
4. systém zobrazí detail uživatele s aktuálními informacemi

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

Formulář pro editaci uživatele

- email - textfield
- password - textfield
- password confirmation - textfield
- admin? - checkbox
- update - button

5.3.2.5 Správa veřejného pdf viz obrázek 9

Aktéři Admin

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému a má práva administrátora

Záměr Administrátor změní vlastnosti veř. pdf a následně jej smaže

Hlavní scénař

1. uživatel vyhledá ve veřejném repozitáři pdf, které chce editovat
2. uživatel zvolí možnost editovat dané pdf
3. systém zobrazí formulář pro editaci veřejného pdf
4. uživatel zadá nové jméno, popis pdf a přiřadí mu tagy, ke kterým má být veř. pdf přiděleno a odešle formulář
5. systém zobrazí veřejný repozitář pdf se změnenými vlastnostmi daného pdf
6. uživatel zvolí u daného pdf možnost smazat pdf
7. systém zobrazí zprávu, zda si uživatel opravdu přeje pdf smazat
8. uživatel potvrdí úmysl smazat veřejné pdf
9. systém zobrazí veřejný repozitář pdf bez daného pdf

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

Formulář pro editaci veřejného pdf

- name - textfield
- description - textfield
- tags - textfield
- update - button

5.3.2.6 Správa veřejné šablony viz obrázek 9

Aktéři Admin

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému a má práva administrátora

Záměr Administrátor změní vlastnosti veř. šablony a následně jí smaže

Hlavní scénař

1. uživatel vyhledá ve veřejném repozitáři šablon tu, kterou chce editovat
2. uživatel zvolí možnost editovat danou šablonu
3. systém zobrazí formulář pro editaci veřejnéhé šablony
4. uživatel zadá nové jméno, popis šablony a přiřadí jí tagy, ke kterým má být veř. šablona přidělena a odešle formulář
5. systém zobrazí veřejný repozitář šablon se změnenými vlastnostmi dané šablony

6. uživatel zvolí u dané šablony možnost smazat šablonu
7. systém zobrazí zprávu, zda si uživatel opravdu přeje šablonu smazat
8. uživatel potvrdí úmysl smazat veřejnou šablonu
9. systém zobrazí veřejný repozitář šablon bez dané šablony

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

Formulář pro editaci veřejné šablony

- name - textfield
- description - textfield
- tags - textfield
- update - button

5.3.2.7 Vytvoření pdf viz obrázek 11

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel vytvoří pdf

Hlavní scénař

1. uživatel zvolí možnost vytvoření nového pdf
2. systém zobrazí formulář pro nové pdf
3. uživatel vyplní jméno a popis pdf
4. uživatel vybere pdf soubor s formulářem jako přílohu a odešle formulář
5. systém zobrazí detail nového pdf, kde bude zobrazen pdf formulář a k němu vygenerované pdf

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

4a systém zkontroluje, zda je příloha typu pdf

5a vygenerované pdf je nahraný pdf formulář s popisky jmen u fieldů

Formulář pro vytvoření pdf

- name - textfield
- description - textarea
- attachment - filefield

5.3.2.8 Vytvoření fieldu viz obrázek 12

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel vytvoří field pro šablonu

Hlavní scénař

1. uživatel zvolí možnost vytvoření nového fieldu v šabloně
2. systém zobrazí formulář pro vytvoření fieldu
3. uživatel zadá jméno, popis, originální jméno fieldu z formuláře, výchozí hodnotu, typ fieldu, nastaví validace a odešle formulář
4. systém zobrazí seznam fieldů pro šablonu obsahující právě vytvořený field

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

2a systém zobrazí v seznamu originálních fieldů z formuláře jen ty, které ještě nejsou přiřazeny v jiných fieldech

Formulář pro vytvoření fieldu

- name - textfield
- description - textarea
- original name - selectbox
- type - selectbox
- default value - textarea, textfield, radiobutton, checkbox, selectbox (podle zvoleného typu fieldu - type)
- validation - checkbox
- error message - textfield

5.3.2.9 Vytvoření kalkulace viz obrázek 12

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel vytvoří kalkulaci pro šablonu

Hlavní scénař

1. uživatel zvolí možnost vytvořit kalkulaci v šabloně
2. systém zobrazí formulář pro kalkulaci
3. uživatel zadá jméno kalkulace
4. uživatel zadá matematickou kalkulaci skládající se ze jmen fieldů a matematických symbolů
5. pokud se uživatel rozhodne, že je hotov, zvolí možnost kontroly kalkulace
6. systém po kontrole kalkulace zobrazí možnost uložení kalkulace
7. uživatel zvolí možnost uložení kalkulace
8. systém zobrazí šablonu se seznamem kalkulací obsahujícím právě vytvořenou kalkulaci

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

4a systém zkontroluje syntaxi zadané kalkulace dle nadefinované gramatiky

Formulář pro vytvoření kalkulace

- name - textfield
- calculation term - textfield
- check calculating term - button
- create - button

5.3.2.10 Vytvoření skupiny viz obrázek 12

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel vytvoří skupinu polí pro šablonu

Hlavní scénař

1. uživatel vyplní jméno skupiny ve fromuláři pro přidání skupiny v šabloně a odešle
2. systém zobrazí šablonu s přidanou skupinou
3. uživatel zvolí u vytvořené skupiny z možnosti fieldů, ten který chce do skupiny přidat a zvolí přidat
4. systém zobrazí šablonu se skupinou obsahující field, který uživatel přidal

   kroky 3 a 4 uživatel opakuje, dokud se nerozhodne, že je hotov
5. uživatel zvolí pomocí šipky nahoru nebo dolů posunutí fildu/skupiny mezi fieldy ve skupinách/skupinami
6. systém zobrazí šablonu s novým umístěním fieldu/skupiny

   kroky 5 a 6 uživatel opakuje, dokud se nerozhodne, že je hotov

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

1a systém validuje, zda jméno skupiny již neexistuje v šabloně

3a systém nabídne uživateli fieldy, které jestě nejsou vloženy v některé skupině v rámci šablony

Formulář pro vytvoření skupiny

- name - textfield
- create - button

5.3.2.11 Vytvoření šablony viz obrázek 12

≪ jméno případu užití ≫ - odkaz na jiný případ užití

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel vytvoří šablonu pro pdf a kalkulace v šabloně

Hlavní scénař

1. uživatel zvolí možnost vytvoření šablony
2. systém zobrazí seznam pdf uživatele
3. uživatel zvolí pdf, ke kterému chce šablonu vytvořit
4. systém zobrazí formulář pro zadání jména šablony
5. uživatel zadá jméno šablony a odešle formulář
6. systém zobrazí seznam vytvořených fieldů v šabloně a možnost vytvořit nový field
7. ≪ vytvoření fieldu ≫

   krok 7 uživatel opakuje dokud se nerozhodne, že je hotov
8. uživatel zvolí možnost pokračovat ve vytváření šablony
9. systém zobrazí seznam skupin v šabloně s možností přidat skupinu
10. ≪ vytvoření skupiny ≫

    krok 10 uživatel opakuje, dokud se nerozhodne, že je hotov
11. uživatel zvolí možnost pokračovat ve vytváření šablony
12. systém zobrazí detail hotové šablony s možností vytváření kalkulací
13. ≪ vytvoření kalkulace ≫

    krok 13 uživatel opakuje, dokud se nerozhodne, že je hotov

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

Formulář pro pojmenování šablony

- name - textfield
- create - button

5.3.2.12 Vytvoření dokumentu viz obrázek 11

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel vytvoří dokument dle šablony

Hlavní scénař

1. uživatel zvolí šablonu v seznamu šablon, ze které chce dokument vytvořit
2. systém zobrazí zvolenou šablonu
3. uživatel vyplní šablonu a zvolí možnost vytvoření dokumentu
4. systém zobrazí formulář pro zadání jména dokumentu
5. uživatel zadá jméno dokumentu do formuláře a odešle
6. systém zobrazí detail dokumentu obsahující vygenerované pdf k šabloně s vyplněnými daty dokumentu

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

3a systém dopočítá a vyplní kalkulace, které jsou k šabloně vytvořené

3b systém zkontroluje validace, které jsou nastaveny v šabloně u jednotlivých polí

1. pokud některá z validací neprojde, systém zobrazí chybné hlášení neprošlých validací a přejde ke kroku 2

Formulář pro pojmenování dokumentu

- name - textfield
- create - button

5.3.2.13 Forknutí privátního pdf viz obrázek 10

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel forkne svoje pdf do veřejného repozitáře pdf

Hlavní scénař

1. uživatel vyhledá v seznamu svých pdf to, které si přeje zkopírovat do veřejného repozitáře pdf
2. uživatel zvolí možnost forknout toto pdf
3. systém zobrazí formulář pro forkování pdf do veřejného repozitáře pdf
4. uživatel vyplní jméno, popis pdf a zvolí tagy, ke kterým má být pdf přiděleno
5. uživatel odešle formulář
6. systém zkopíruje pdf do veřejného repozitáře pdf

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

6a systém zkontroluje, zda pdf neexistuje ve veřejném repozitáři pdf

1. pokud pdf existuje ve veř. repozitáři pdf, systém ukončí tento případ užití a zobrazí hlášení o existenci pdf ve veř. repozitáři

Formulář pro forkování pdf

- name - textfield
- description - textarea
- tags - textfield
- fork - button

5.3.2.14 Forknutí privátní šablony viz obrázek 10

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel forkne svojí šablonu do veřejného repozitáře šablon

Hlavní scénař

1. uživatel vyhledá v seznamu svých šablon tu, kterou si přeje zkopírovat do veřejného repozitáře šablon
2. uživatel zvolí možnost forknout tuto šablonu

3. systém zobrazí formulář pro forkování šablon do veřejného repozitáře šablon
4. uživatel vyplní jméno, popis šablony a zvolí tagy, ke kterým má být šablona přidělena
5. uživatel odešle formulář
6. systém zkopíruje šablonu do veřejného repozitáře šablon

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

6a systém zkontroluje, zda šablona neexistuje ve veřejném repozitáři šablon

1. pokud šablona existuje ve veř. repozitáři šablon, systém ukončí tento případ užití a zobrazí hlášení o existenci pdf ve veř. repozitáři

Formulář pro forkování pdf

- name - textfield
- description - textarea
- tags - textfield
- fork - button

5.3.2.15 Forknutí veřejného pdf viz obrázek 10

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel forkne veřejné pdf do svého seznamu pdf

Hlavní scénař

1. uživatel vyhledá ve veřejném repozitáři pdf to, které si chce nahrát do seznamu vlastních pdf
2. uživatel zvolí možnost forknutí veřejného pdf
3. systém zkopíruje pdf do seznamu uživatelových pdf
4. systém zobrazí seznam uživatelových pdf s daným pdf

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

3a systém zkontroluje, zda pdf neexistuje v seznamu uživatelových pdf

1. pokud pdf existuje v seznamu uživatelových pdf, systém ukončí tento případ užití a zobrazí hlášení o existenci pdf v seznamu pdf

5.3.2.16 Forknutí veřejné šablony viz obrázek 10

Aktéři Admin, registrovaný uživatel

Úroveň Uživatelský cíl

Rozsah PDF portál

Vstupní podmínky Uživatel je přihlášen do systému

Záměr Uživatel forkne veřejnou šablonu do svého seznamu šablon

Hlavní scénař

1. uživatel vyhledá ve veřejném repozitáři šablonu, kterou si chce nahrát do seznamu vlastních šablon
2. uživatel zvolí možnost forknutí veřejné šablony
3. systém zkopíruje šablonu do seznamu uživatelových šablon
4. systém zobrazí seznam uživatelových šablon s danou šablonou

Rozšíření

***a** pokud je uživatel déle jak 15 minut neaktivní, bude odhlášen ze systému

3a systém zkontroluje, zda šablona neexistuje v seznamu uživatelových šablon

1. pokud šablona existuje v seznamu uživatelových šablon, systém ukončí tento případ užití a zobrazí hlášení o existenci šablony v seznamu šablon

6 Návrh Implementace

Dalším krokem při vývoji PDF portálu byl návrh implementace. V této kapitole jsem řešil problematiku technického a programového vybavení, otázku grafického vzhledu PDF portálu, testy, gramatiky atd. Přípravná fáze celé implementace výrazně zefektivnila následnou implementaci.

6.1 Gramatika kalkulací

Kalkulace pro šablony mají mít jednoduchý tvar matematických vzorců. Pro kontrolu správného tvaru jsem si vytvořil jednoduchou bezkontextovou gramatiku popisující tyto kalkulace. Dle definece 6.1 (uvedené v literatuře [9]) vypadá gramatika následně. Pro jednodušší vyjádření množiny všech slov a desetinných čísel u neterminálů „WORD“ a „NUMBER“ jsem v gramatice použil regulárního vyjádření. Neterminál „WORD“ reprezentuje field z šablony.

Definice 6.1 *Bezkontextová gramatika je definována jako uspořádaná čtveřice* $G = (\Pi, \Sigma, S, P)$*, kde*

- Π *je konečná množina neterminálních symbolů (neterminálů)*
- Σ *je konečná množina terminálních symbolů (terminálů), přičemž* $\Pi \cap \Sigma = \emptyset$
- $S \in \Pi$ *je počáteční (startovací) neterminál*
- *P je konečná množina pravidel typu* $A \rightarrow \beta$*, kde*
  - *A je neterminál, tedy* $A \in \Pi$
  - β *je řetězec složený z terminálů a neterminálů, tedy* $\beta \in (\Pi \cup \Sigma)^*$

Gramatika pro kalkulace:

$S = S$
$\Sigma = \{+, -, *, /, (,), [0-9], , , [a-zA-Z]\}$
$\Pi = \{S, E, W, WORD, NUMBER\}$
$P = \{$
$S \longrightarrow WORD = E,$
$E \longrightarrow E + E \mid E - E \mid E * E \mid E/E \mid (E) \mid W,$
$W \longrightarrow NUMBER \mid WORD,$
$WORD \longrightarrow (+ \mid -)?([a-zA-Z] + (\backslash\ )?)+,$
$NUMBER \longrightarrow (+ \mid -)?[0-9]+, [0-9]+ \mid (+ \mid -)?[0-9]+$
$\}$

6.2 Softwarové vybavení

Serverová část programového vybavení:

- Operační systém - Linux Debian (Ubuntu distr. 32bit)

- Databázový server - MySQL 5.0
- Web server - Apache2.2 Phusion Passanger 2
- Skriptovací jazyk - Ruby 1.8.6

Klientská část programového vybavení:

- Operační systém - Windows 95/98/Me/NT/2000/XP/7, Unix/Linux, MacOS
- Internetový prohlížeč - Internet Exploler 5.x a vyšší, Mozilla Firefox 3.x.x a vyšší

6.3 Hardwarové vybavení

Hardwarové nároky nejsou příliš velké. Vyhledem k tomu, že jde o internetovou aplikaci běžící na serverové straně, stačí počítač, na němž lze spustit internetový prohlížeč potřebné verze (viz kapitola 6.2)

6.4 Grafický návrh

Další částí návrhu implementace byla grafická podpora. Důležitou otázkou bylo rozmístění prvků v rámci PDF portálu.

6.4.0.17 Přihlašovací stránka

Viz obrázek 4

6.4.0.18 Uživatelské rozhraní

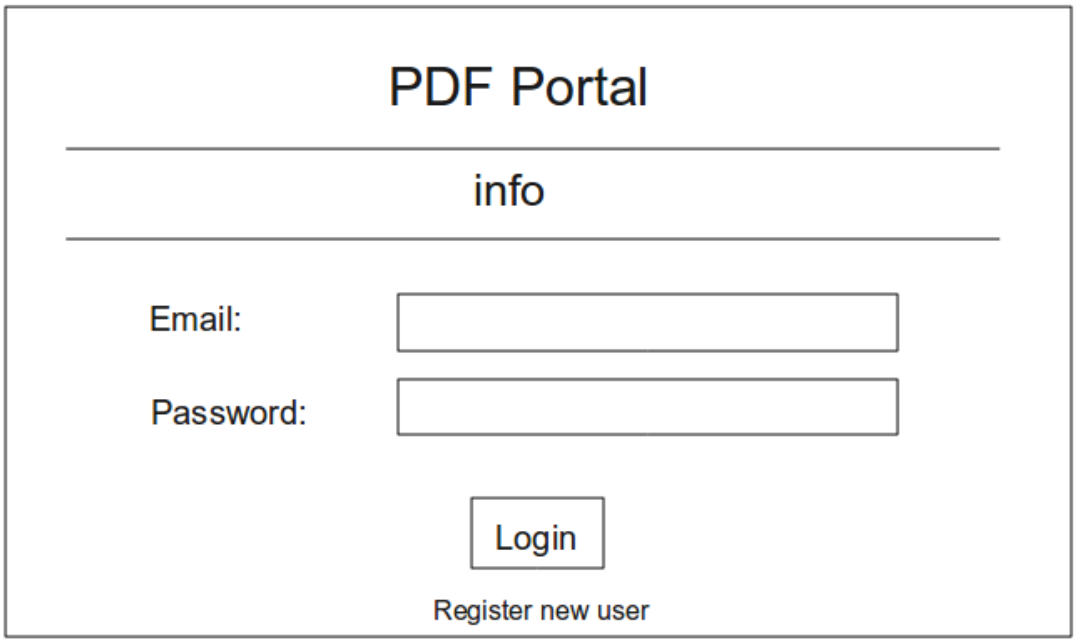
Viz obrázek 5

6.4.0.19 Šablona

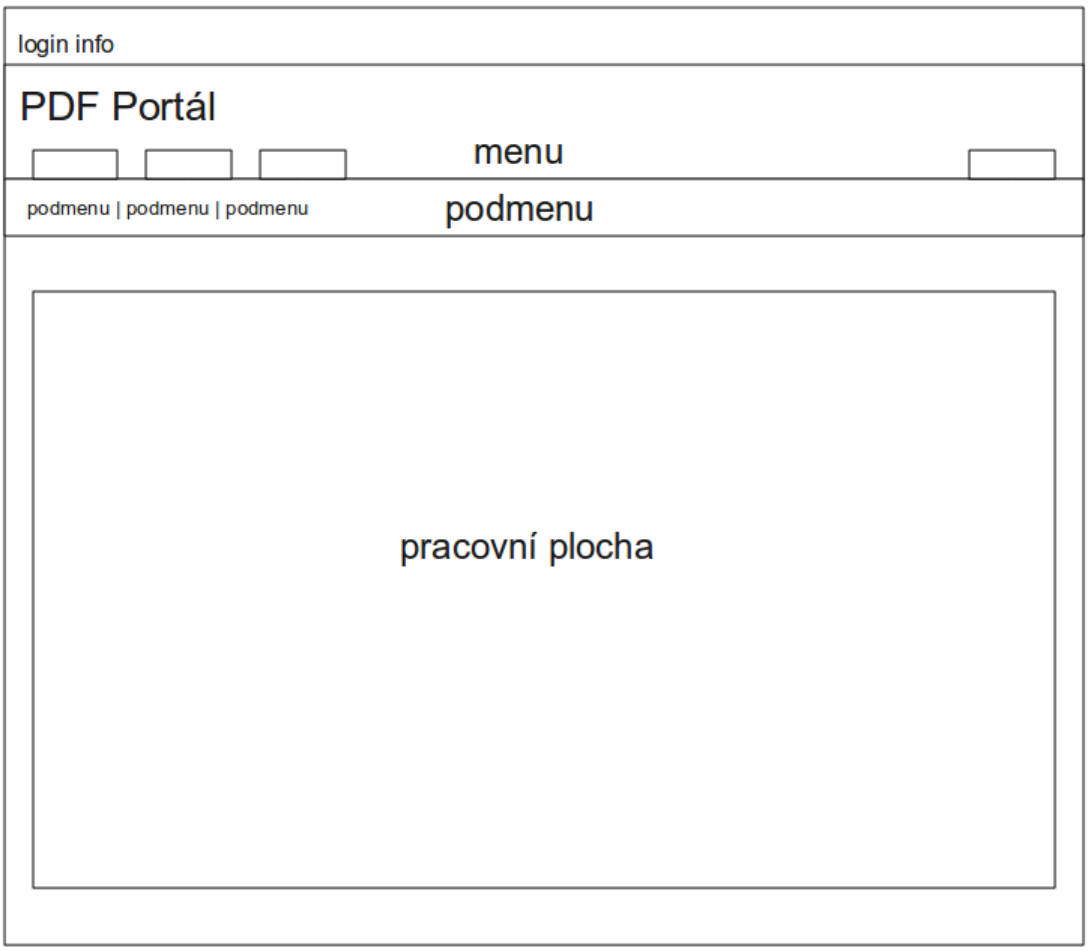
Viz obrázek 6

6.4.0.20 Veřejný repozitář

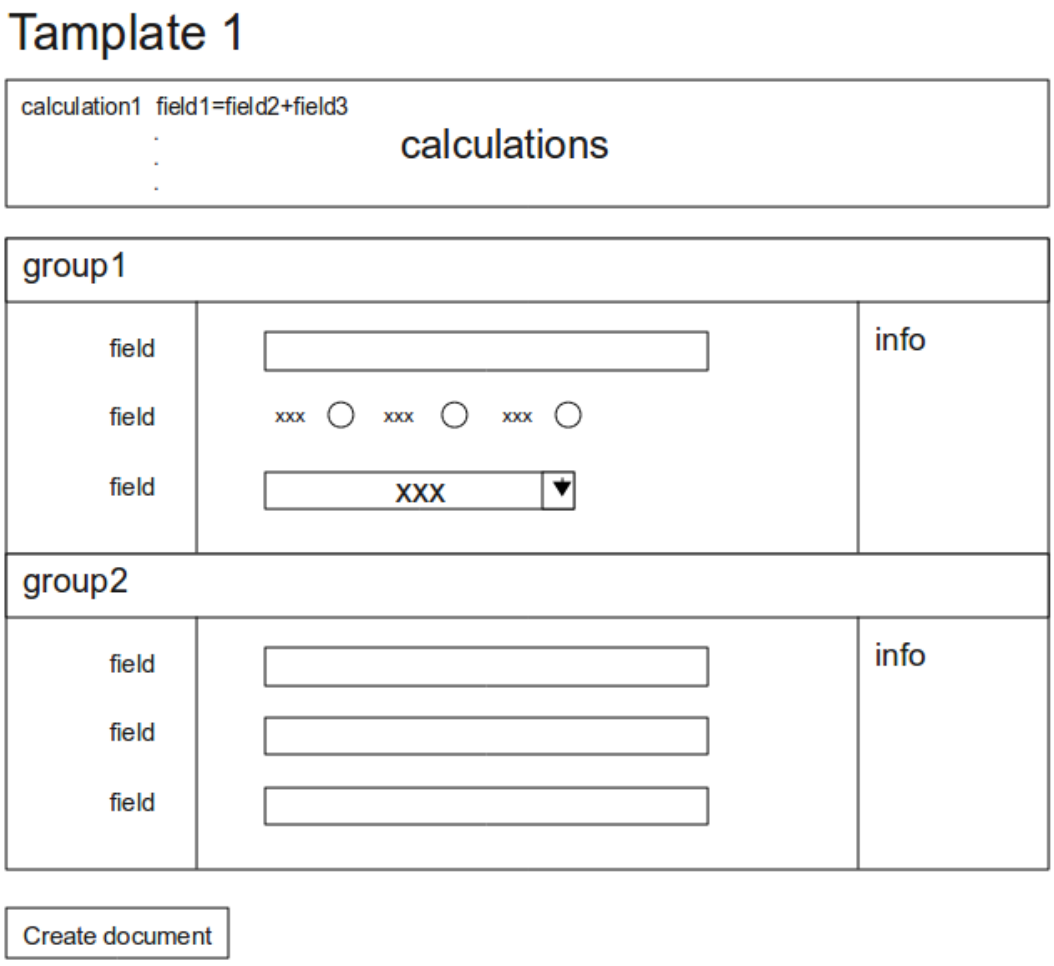
Viz obrázek 7



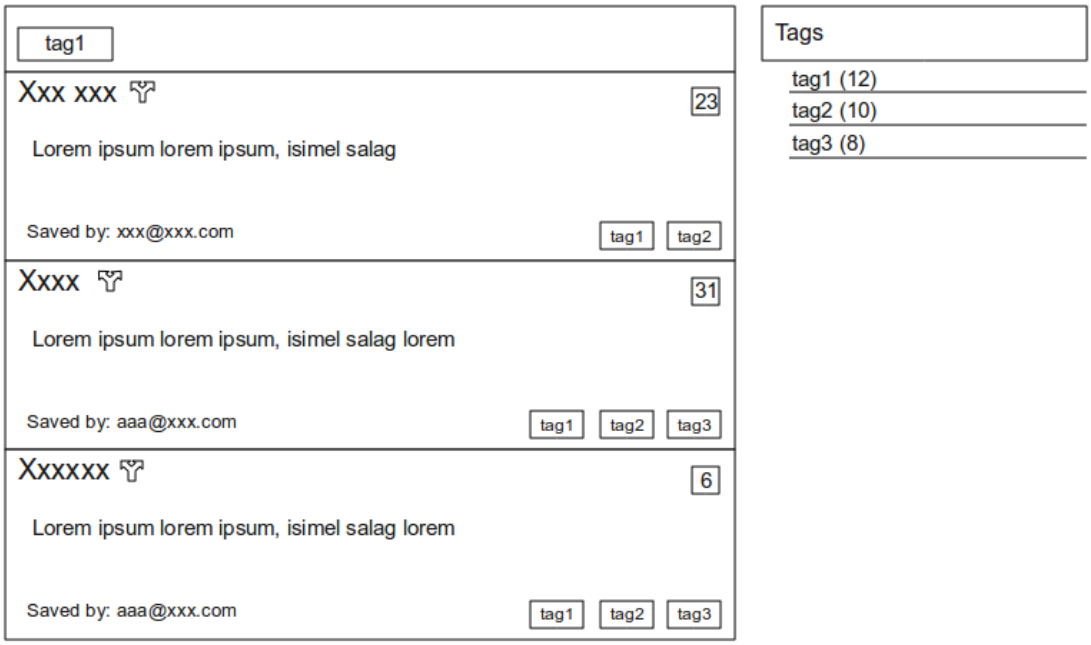
Obrázek 4: Přihlašovací stránka



Obrázek 5: Uživatelské rozhraní



Obrázek 6: Šablona



Obrázek 7: Veřejný repozitář(pdf nebo šablon)

7 Implementace

Podrobné informace o implementaci diplomové práce jsou obsaženy na přiloženém CD. Uživatelská, aministrátorská a programátorská příručka jsou v adresáři „prirucky/“ na CD ve formě videa. Přiložené CD taktéž obsahuje kompletní zdrojové kódy celé aplikace.

7.1 Použité programové prostředky

Vývoj aplikace:

- operační systém - Linux Ubuntu 9.10 (Karmic Koala)
- databáze (relační) - MySql 5.1.37-1Ubuntu5.1
- databáze (dokumentová) - CouchDB 0.9.1
- databáze (systémové fronty pro workery) - Redis 1.3.2
- skriptovací jazyk - Ruby 1.8.7 (2009-06-12 patchlevel 174)
- skriptovací jazyk (java) - JRuby 1.4.0 (ruby 1.8.7 patchlevel 174)
- framework - Ruby on Rails 2.3.5
- vývojové prostředí - Gedit 2.28.0
- grafický editor - Gimp 2.6
- subversion systém - GitHub

Pro sestavení dokumentace:

- sázecí systém - LaTeX
- textový editor - TeXMaker 1.9.2
- tvorba diagramů - CaseComplete
- tvorba mediálních příruček - Camtasia Studio

Pro demoverzi na internetu:

- operační systém - Linux Debian (Ubuntu distr. 32bit)
- webserver - Apache + Phusion Passanger 2
- skriptovací jazyk - ruby 1.8.6 (2008-08-08 patchlevel 286)
- skriptovací jazyk (java) - jruby 1.4.0 (ruby 1.8.7 patchlevel 174)
- databázový server (relační) - MySql 5.0
- databázový server (dokumentový) - CouchDB 0.9.2
- databázový server (systémové fronty pro workery) - Redis 1.3.2

7.2 Parser pro gramatiku

Pro implementaci parseru na uvedenou gramatiku k jednoduchým kalkulacím v kapitole 6.1, jsem použil knihovnu „Treetop“. Treetop je knihovna, která umožňuje jednoduše implementovat interpretr pro vámi nadefinovaný jazyk jak po sémantické, tak po syntaktické stránce . Po nadefinování gramatiky, Treetop vygeneruje parser, který transformuje vstupní řetězec do vašeho jazyka v abstraktním stromu syntaxe, reprezentujícího jeho strukturu. Pokud vstupní řetězec neodpovídá syntaxi nadefinovaného jazyka v gramatice, parser vrátí nulovou hodnotu místo abstraktního stromu. Této vlastnosti jsem využil ke kontrole správné syntaxe pro kalkulace, které budou uživatelé zadávat do systému.

Jelikož Treetop nepodporuje gramatiky, ve kterých se vyskytuje levá rekurze, musel jsem upravit navrhovanou gramatiku tak, aby byla bez levé rekurze. Vypadá následovně.

$$
\begin{aligned}
&S = S \\
&\Sigma = \{+, -, *, /, (,), [0-9], , , [a-zA-Z]\} \\
&\Pi = \{S, E, EREST, W, WORD, NUMBER\} \\
&P = \{ \\
&\quad S \longrightarrow WORD = E, \\
&\quad E \longrightarrow (E)EREST \mid WEREST, \\
&\quad EREST \longrightarrow +EEREST \mid -EEREST \mid *EEREST \mid /EEREST \\
&\quad W \longrightarrow NUMBER \mid WORD, \\
&\quad WORD \longrightarrow (+ \mid -)?([a-zA-Z] + (\backslash\ )?)+, \\
&\quad NUMBER \longrightarrow (+ \mid -)?[0-9]+, [0-9]+ \mid (+ \mid -)?[0-9]+ \\
&\}
\end{aligned}
$$

Pro nadefinování gramatiky v Treetop používám tatkéž regulární výrazy. Nadefinování gramatiky v Treetop vypadá následovně.

```
grammar CalcGrammar

  rule S
    WORD '=' E
  end

  rule E
    W ERest / '(' E ')' ERest
  end

  rule ERest
    '+' (E ERest) / '−' (E ERest) / '∗' (E ERest) / '/' (E ERest) / ''
  end

  rule W
    NUMBER / WORD
  end

  rule NUMBER
```

```
    (( '+' / '-')? [0-9]+ '.' [0-9]+) /
    ('+' / '-')? [\d]+
  end

  rule WORD
    ('+' / '-')? '#{' [(\w+( )?)]+ '}'
  end

end
```

Výpis 5: Gramatika v Treetop

7.3 Testování a podpora WWW prohížečů

Aplikace podporuje standart XHTML 1.0 a CSS 2.0. Při vývoji jsem používal prohlížeč Mozilla Firefox. Aplikaci jsem otestoval dále na prohlížečích Internet Exploler a Google Chrome. U Google Chrome prohlížeče jsem narazil na absenci zobrazování PDF souborů jako součást html.

7.4 Umístění demonstrační verze

Demonstrační verze je umístěna na adrese „http://simpleforms.org“.

Přístupy do systému:

admin :

email - nevralaz@gmail.com

heslo - admin

7.5 Design stránek

Pro návrh vzhledu stránek byly využity kaskádové styly, které byly testovány prostřednictvím prohlížeče Mozilla Firefox. Kaskádové styly jsou uloženy v souboru „public/stylesheets/“, jejich výhodou je snadná editace celkového vzhledu stránek.

8 Testy

V posledních letech začali vývojáři testovat aplikace na stále nižších úrovních. Současný rámec dynamických jazyků, které nenabízejí kompilátory pro základní zachytávání chyb, dělají testování více důležitějším. Testování by mělo být základním pilířem vývoje všech projektů. Zaručujeme tím určitou kvalitu softwaru. Níže je základní seznam výhod testování.

- Testování zlepší vaše návrhy.
- Testování snižuje prezenci zbytečných kódů. Pokud píšete první testy, obvykle pak píšete jen tolik kódu, aby prošel testy. Nesmíme opomenout to, že testování určuje kvalitu softwaru.
- Testově řízený vývoj je více motivující. Každý modelový případ vytváří menší problémy. Řešení těchto problémů je přínosné a motivující. Jestliže děláte testově řízený vývoj, hodiny rychleji utíkají.
- Testování umožňuje větší volnost. Pokud máme testovací případy, které pravděpodobně zachytí chyby, zjistíte, že jste více ochotni provádět zlepšení kódu.

V mé diplomové práci jsem k testování použil nástroje RSpec a Cucumber viz níže.

8.1 RSpec

RSpec je Ruby knihovna, která umožňuje vytvořit specifikaci softwaru při vývoji. Pomáhá vyvíjet software za pomoci chováním-řízeného vývoje (behavior-driven development BDD) a testově-řízeného vývoje(test-driven development TDD). Je to skvělý nástroj pro testování softwaru. Vyznačuje se velmi čitelnou syntaxí a může také sloužit jako technická dokumentace kódu. RSpec navíc umožňuje vytvářet pro testování „mock objekty“, které umožňují simulovat chování objektů v testech a kontrolovat je. Jde například o objekty, které komunikují s webovými službami atd...

Níže uvedu základní kroky testově řízeného vývoje pomocí RSpec.

- Napíšete testy popisují chování konkrétní části systému (softwaru).
- Spustíte testy. Testy samozřejmě selžou, protože jste zatím nenaimplementovali testované části systému. Toto je důležitý krok testování v testovacích případech. Ujistěte se, že testy selžou v případech, ve kterých by měly.
- Naimplementujte tolik kódu, aby testy prošly.
- Spusťte testy a ověřte zda testy prošly.

Ve výpisu 6 je zobrazen kousek testu popisující model Pdf při uploadování. V testu kontroluji validace při uploadování nového pdf do systému. Jak je vidět, test je čitelný a lze z něj krásně vyčíst chování pdf při uploadování. Všechny RSpec testy naleznete v adresáři „spec/models“. Další informace o RSpec testech naleznete na domovských WWW stránkách [10].

```
describe Pdf, "Upload" do

  it "should raise validates errors" do
    pdf = Pdf.new()
    pdf.should_not be_valid
    pdf.errors.values.should have(2).items
    pdf.errors.on(:name).should have(1).items
    pdf.name = 'test'
    pdf.should_not be_valid
    pdf.errors.values.should have(1).items
    pdf.errors.on(:name).should be_nil
    pdf.name = ''
    pdf.should_receive(:valid_attachment?).and_return(true)
    pdf.should_not be_valid
    pdf.errors.on(:name).should have(1).items
  end

end
```

Výpis 6: Ukázka RSpec testu

8.2 Cucumber

Dalším z velmi účinných nástrojů pro testování je Cucumber (taktéž BDD a TDD). Jde o nejnovější přírustek do rodiny RSpec nástrojů.

Cucumber je navržen tak, aby umožnil spustit dokument se specifikací napsanou ve formě prostého textu (známe jako „příběhy“). Jelikož cucumber příběhy jsou velmi podobné s případy užití, které se většinou definují ve fázi návrhu systému, dají se využít k testování těchto případů užití.

Ve výpisu 7 můžeme vidět ukázku dokumentu s příběhy pro manažování tagů v systému. Background část kódu se spustí před každým scénařem (příběhem). Z výpisu je vidět, jak je Cucumber čitelný a jak jednoduše se z něj dá vyčíst chování systému.

```
Feature: Manage Tags
  In order to make a tag
  As an admin
  I want to create and manage tags

  Background:
    When I go to path "/access/login"
    And I fill in the following:
```

```
        |email    |testuser@gmail.com|
        |password|testuser           |
    And I press "Login"
    Then I should see "Logged in user: testuser@gmail.com"

  Scenario: Tags List
    Given the following tag records
      |name    |description       |
      |Cucumber|desc of cucumber|
      |Tests    |desc of tests    |
    When I go to path "/tags"
    Then I should see "Cucumber"
    And I should see "Tests"

  Scenario: Update Tags
    When I go to path "/tags"
    Then I should see "Cucumber"
    When I visit tag page "edit" for "Cucumber"
    Then I should see "Edit Tag"
    And the "Name" field should contain "Cucumber"
    When I fill in "Name" with "NewTag"
    And press "Update"
    Then I should see "NewTag"
    And I should not see "Cucumber"

end
```

Výpis 7: Ukázka Cucumber příběhů

Z výpisu je také vidět, že se příběhy skládájí z kroků začínajících na slova „Given“, „When“, „Then“, „And“. Chování těchto kroků se k scénářům musí dodefinovat, pokud již nejsou defaultně nadefinované v souboru „features/step_definitions/web_steps.rb“. Například definici kroku pro Pdf features „When I visit tag page "edit" for "Cucumber"“ můžeme vidět ve výpisu 8. Definici určujeme podle počátečního slova a regulárního výrazu, který následuje. Regulární výraz vyfiltruje atributy, které můžeme v příbězích měnit. Uvnitř definice naimplementujeme kód, který se má provést při volání tohoto kroku.

```
When /^I visit tag page "(.+)" for "(.+)"$/ do |action, tag_name|
  tag = Tag.by_name(:key => tag_name).first
  visit url_for (: controller => 'tags', :action => action, :id => tag.id)
end
```

Výpis 8: Definice kroku pro Cucumber (z „features/step_definitions/manage_tags.rb“)

Cucumber příběhy naleznete v adresáři aplikace „features/“. Více informací o Cucumber testech naleznete na WWW stránkách [11].

9 Závěr

V práci jsem provedl kompletní analýzu portálu interaktivních formulářů a vytvořil tak návrh aplikace. Na základě dostupných technologií jsem realizoval tento portál.

9.1 Teoretický závěr

Seznámil jsem se podrobně s problematikou PDF formulářů a prací s nimi. Osvojil si práci s dokumentově orientovanou databází a vyzkoušel nové typy testování webových aplikací. Dále jsem si rozšířil obzory v oblasti spravování úkolů pro systém/server a jeho zatěžování pomocí systémových front s úkoly a workery. V práci jsem si také osvěžil znalosti z teoretické informatiky při vytváření gramatiky pro kalkulace.

9.2 Implementační závěr

Důkladnost provádění jednotlivých kroků analýzy datové i funkční se výrazným způsobem projevila při návrhu implementace a samotné implementaci projektu. Čas věnovaný návrhu při implementaci se mnohonásobně vrátil. Jediným větším problémem, se kterým jsem se při implementaci setkal, byla absence nekomerečních knihoven pro práci s PDF formuláři a zvolení vhodného workflow pro práci s těmito formuláři. Dalšímu rozvoji tohoto projektu bych se chtěl nadále věnovat.

10 Reference

[1] KREJČÍ, Richard. PDF formuláře: Vyplňování, odesílání a zpracovávání formulářových dat. *Grafika : PDF* [online]. 12.06.2002, [cit. 2010-04-27]. *D*ostupný z WWW: <http://www.grafika.cz/art/pdf/pdfform6.html>.

[2] MANNOVÁ, Božena. *Technologie pro e-vzdělávání* [online]. Praha, 2005. 75 s. Seminární práce. ČVUT, VSB-TUO, Slovenská technická universita Bratislava, Ústav informatiky a softvérového inžinierstva, Czech ACM Chapter. Dostupné z WWW: <http://acm.vsb.cz/tpev2005/TPEV2005-sbornik.pdf>.

[3] KREJČÍ, Richard. PDF formuláře: obecný úvod. *Grafika : PDF* [online]. 25.04.2002, 6, [cit. 2010-04-27]. *D*ostupný z WWW: <http://www.grafika.cz/art/pdf/clanek1953461646.html>.

[4] *PDFlib* [online]. 2009 [cit. 2010-04-27]. Dostupné z WWW: <http://www.pdflib.com/>.

[5] *IText* [online]. 2008 [cit. 2010-04-27]. Dostupné z WWW: <http://itextpdf.com/>.

[6] *PlanetPDF* [online]. 2009 [cit. 2010-04-25]. Dostupné z WWW: <http://www.planetpdf.com/>.

[7] *PDFzone* [online]. 2010 [cit. 2010-04-25]. Dostupné z WWW: <http://www.pdfzone.com/>.

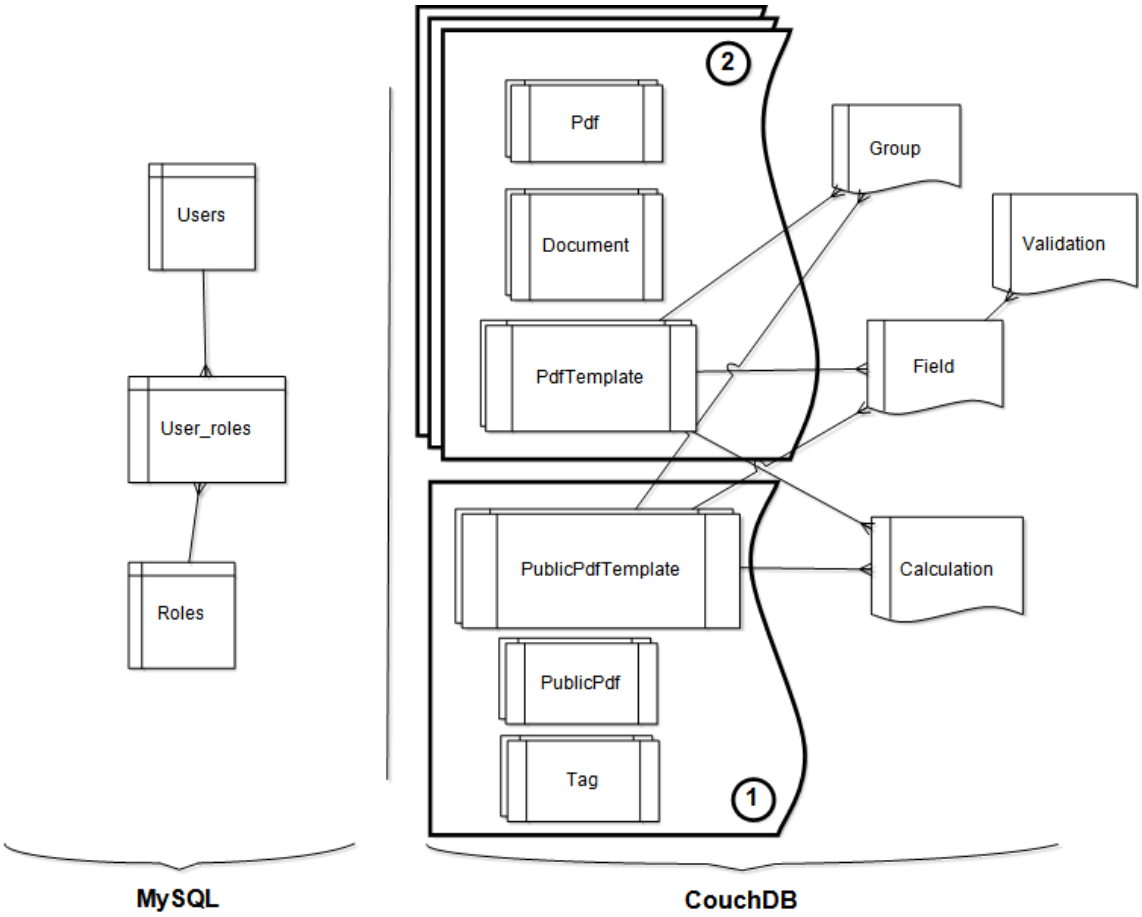
[8] *Apache CouchDB* [online]. 2008 [cit. 2010-04-27]. Dostupné z WWW: <http://couchdb.apache.org/index.html>.

[9] Jančar, Petr. *Teoretická informatika* [online]. Ostrava, 2007. 336 s. Učební text. Vysoká škola báňská – Technická univerzita Ostrava. Dostupné z WWW: <http://www.cs.vsb.cz/jancar/TEORET-INF/ti-text.2010-01-20.pdf>.

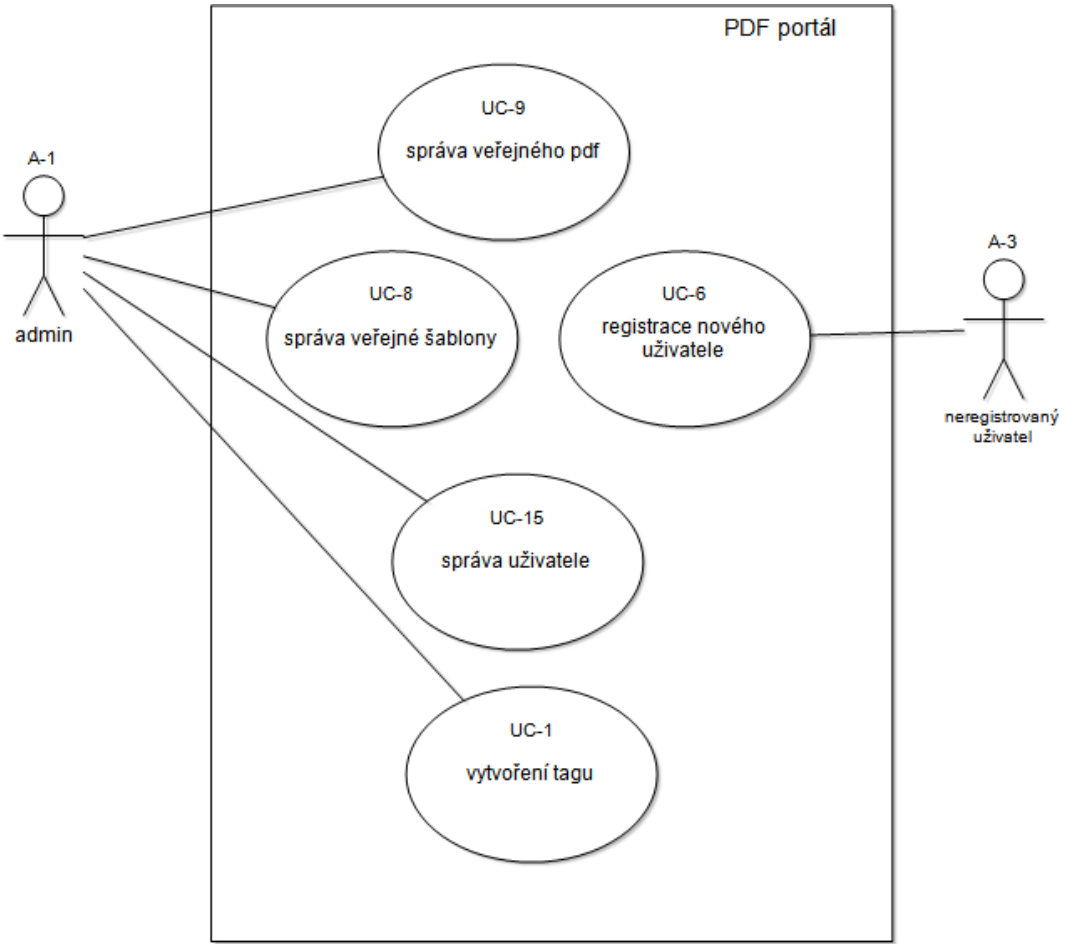
[10] *RSpec* [online]. 2010 [cit. 2010-04-26]. Dostupné z WWW: <http://rspec.info/>.

[11] *Cucumber* [online]. 2010 [cit. 2010-04-26]. Dostupné z WWW: <http://cukes.info/>.

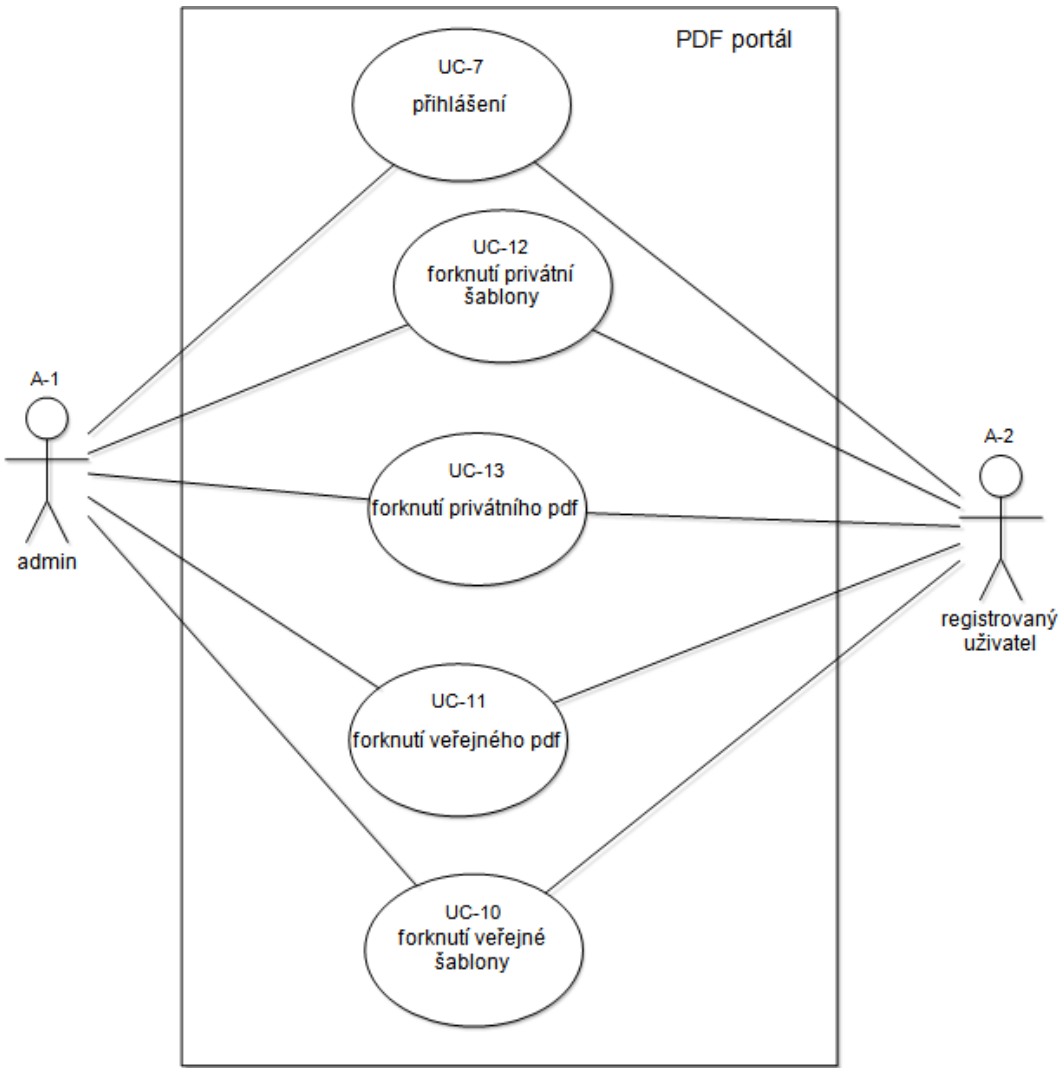
A Grafy



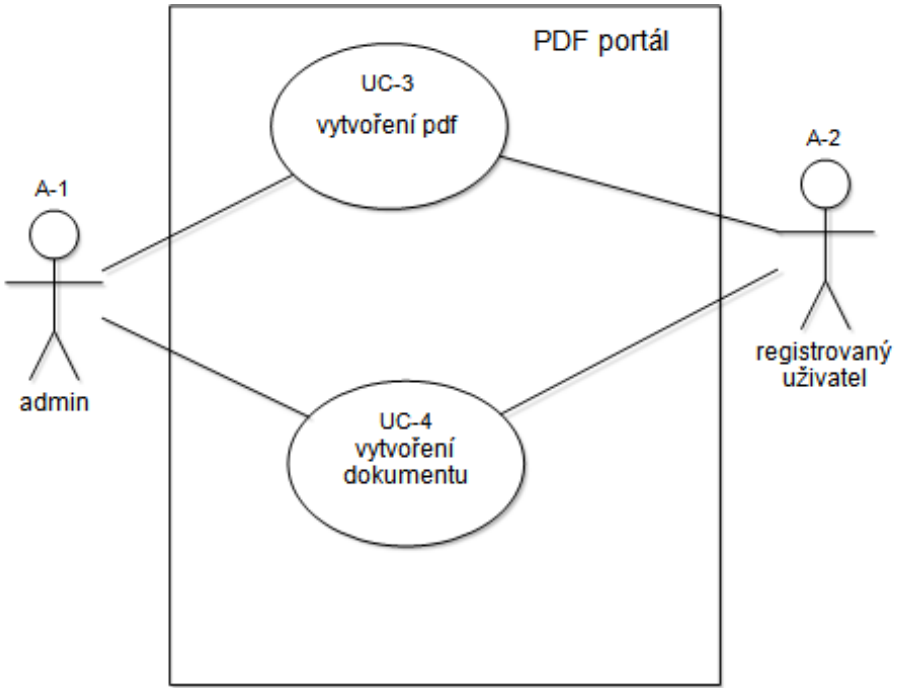
Obrázek 8: Schéma databází



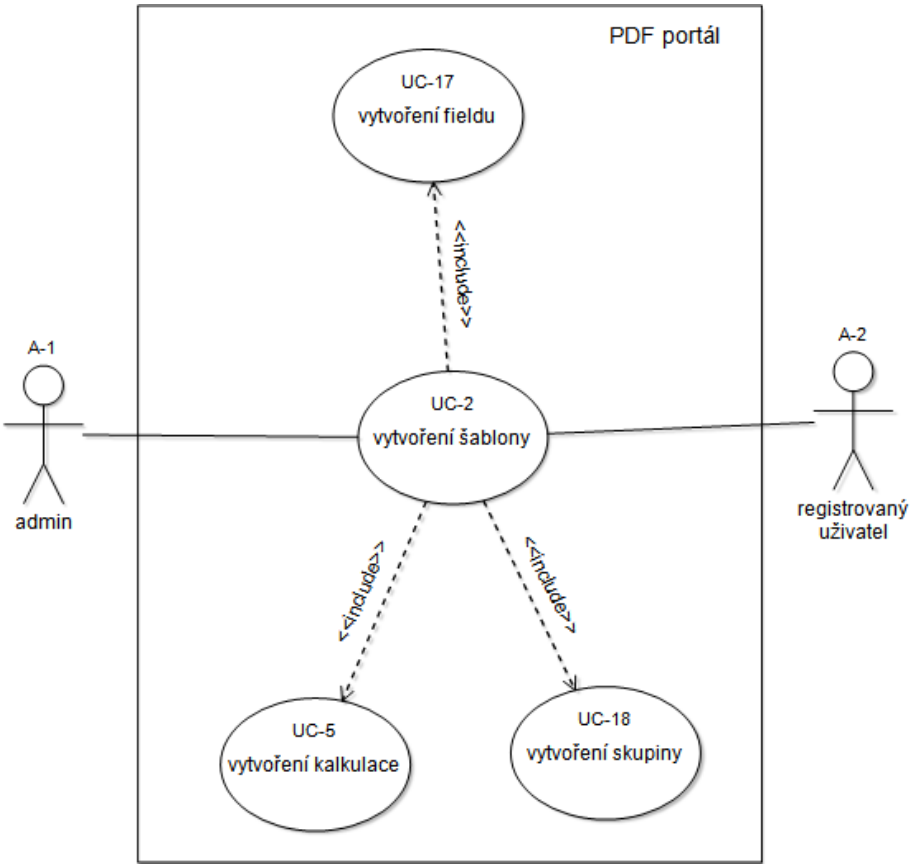
Obrázek 9: Use Case Diagram 1



Obrázek 10: Use Case Diagram 2



Obrázek 11: Use Case Diagram 3



Obrázek 12: Use Case Diagram 4

B Tabulky

Atribut	Dat.typ	Velikost	Klíč	Null	Výchozí	Extra	Český popis
id	integer	11	A	N	Null	auto increment	id uživatele, primární klíč
email	varchar	255	N	A	Null	unikátní	email uživatele, přihlašovací údaj
password_hash	varchar	255	N	A	Null		hashované heslo uživatele
couchdb_name	varchar	255	N	A	Null		jméno uživat. DB pro vlastní repozitář
lock_version	integer	11	N	A	0		atribut pro zamykání entity
created_at	datetime		N	A	Null		datum vytvoření uživatele
updated_at	datetime		N	A	Null		datum poslední úpravy uživatele

Tabulka 2: Tabulka Users

Atribut	Dat.typ	Velikost	Klíč	Null	Výchozí	Extra	Český popis
id	integer	11	A	N	Null	auto increment	id role, primární klíč
name	varchar	255	N	A	Null		jméno role
description	varchar	255	N	A	Null		popis role
created_at	datetime		N	A	Null		datum vytvoření role
updated_at	datetime		N	A	Null		datum poslední úpravy role

Tabulka 3: Tabulka Roles

Atribut	Dat.typ	Velikost	Klíč	Null	Výchozí	Extra	Český popis
id	integer	11	A	N	Null	auto increment	id vazební entity, primární klíč
user_id	integer	11	N	A	Null		id uživatele
role_id	integer	11	N	A	Null		id role
created_at	datetime		N	A	Null		datum vytvoření vazební entity
updated_at	datetime		N	A	Null		datum poslední úpravy vazební entity

Tabulka 4: Tabulka User_roles

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
_id	string	A	N	Null	autogenerate UUID	id pdf, primární klíč
_rev	string	N	N	Null	autogenerate rev. nubmer	číslo revize pdf
_attachments	hash	N	A	Null		hash obsahující pdf soubory
name	string	N	A	Null		jméno pdf
attachment	string	N	A	Null		jméno klíče pro pdf formulář z _attachments
attachment_describe	string	N	A	Null		jméno klíče pro describe pdf z _attachments
couchrest-type	string	N	A	Null		typ dokumentu
description	string	N	A	Null		popis pdf
done	boolean	N	A	false		určení zda je k nahranému pdf vygenerován describe pdf
forked	boolean	N	A	false		určení zda bylo pdf forkováno
forked_pdf_id	string	N	A	Null		id pro forkování pdf
user	string	N	A	Null		email uživatele, který šablonu vytvořil
original_fields	hash	N	A	Null		hash s údaji originálních polí v pdf formuláři
created_at	string	N	A	Null		čas vytvoření pdf
updated_at	string	N	A	Null		čas poslední úpravy pdf

Tabulka 5: Dokument Pdf

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
_id	string	A	N	Null	autogenerate UUID	id pdf, primární klíč
_rev	string	N	N	Null	autogenerate rev. nubmer	číslo revize pdf
_attachments	hash	N	A	Null		hash obsahující pdf soubory
name	string	N	A	Null		jméno pdf dokumentu
attachment	string	N	A	Null		jméno klíče pro pdf formulář z _attachments
attachment_describe	string	N	A	Null		jméno klíče pro describe pdf z _attachments
couchrest-type	string	N	A	Null		typ dokumentu
description	string	N	A	Null		popis pdf
done	boolean	N	A	false		určení zda je k nahranému pdf vygenerován describe pdf
forked	boolean	N	A	false		určení zda bylo pdf forkováno
forked_pdf_id	string	N	A	Null		id pro forkování pdf
forked_no	integer	N	A	0		počet forknutí do privátních repozitářů
user	string	N	A	Null		email uživatele, který šablonu vytvořil
original_fields	hash	N	A	Null		hash s údaji originálních polí v pdf formuláři
tags	array	N	A	Null		pole s id tagů, ke kterým je PublicPdf přiděleno
created_at	string	N	A	Null		čas vytvoření pdf
updated_at	string	N	A	Null		čas poslední úpravy pdf

Tabulka 6: Dokument PublicPdf

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
_id	string	A	N	Null	autogenerate UUID	id pdf šablony, primární klíč
_rev	string	N	N	Null	autogenerate rev. nubmer	číslo revize šablony
calculations	array	N	A	Null		pole obsahující kalkulace(viz. tabulka 12)
name	string	N	A	Null		jméno pdf šablony
couchrest-type	string	N	A	Null		typ dokumentu
description	string	N	A	Null		popis šablony
forked	boolean	N	A	false		určení zda byl pdf dokument forkován
forked_template_id	string	N	A	Null		id pro forkování šablony
user	string	N	A	Null		email uživatele, který šablonu vytvořil
fields	array	N	A	[]		pole obsahující informace o polích v šabloně (viz. tabulka 11)
groups	array	N	A	[]		pole se skupinamy (viz. tabulka 13)
pdf_id	string	N	A	Null		id pdf ke kterému šablona patří
created_at	string	N	A	Null		čas vytvoření šablony
updated_at	string	N	A	Null		čas poslední úpravy šablony

Tabulka 7: Dokument PdfTemplate

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
_id	string	A	N	Null	autogenerate UUID	id pdf šablony, primární klíč
_rev	string	N	N	Null	autogenerate rev. nubmer	číslo revize šablony
calculations	array	N	A	Null		pole obsahující kalkulace(viz. tabulka 12)
name	string	N	A	Null		jméno pdf šablony
couchrest-type	string	N	A	Null		typ dokumentu
description	string	N	A	Null		popis šablony
forked	boolean	N	A	false		určení zda byl pdf dokument forkován
forked_template_id	string	N	A	Null		id pro forkování šablony
forked_no	integer	N	A	0		počet forkování šablony do privátních repoz.
user	string	N	A	Null		email uživatele, který šablonu vytvořil
fields	array	N	A	[]		pole obsahující informace o polích v šabloně (viz. tabulka 11)
groups	array	N	A	[]		pole se skupinamy (viz. tabulka 13)
pdf_id	string	N	A	Null		id pdf ke kterému šablona patří
tags	array	N	A	[]		pole id tagů, ke kterým byla veřejná šablona přidělena
created_at	string	N	A	Null		čas vytvoření šablony
updated_at	string	N	A	Null		čas poslední úpravy šablony

Tabulka 8: Dokument PublicPdfTemplate

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
_id	string	A	N	Null	autogenerate UUID	id dokumentu, primární klíč
_rev	string	N	N	Null	autogenerate rev. nubmer	číslo revize dokumentu
name	string	N	A	Null		jméno dokumentu
_attachments	hash	N	A	Null		příloha (vygenrovaný pdf s vyp. údaji)
couchrest-type	string	N	A	Null		typ dokumentu
done	boolean	N	A	false		příznak zda je pdf k dokumentu vygenrováno
pdf_template_id	string	N	A	Null		id pdf šablony, ke které dokument patří
values	hash	N	A	Null		hash s údaji pro vyplnění orig. pdf
created_at	string	N	A	Null		čas vytvoření dokumentu
updated_at	string	N	A	Null		čas poslední úpravy dokumentu

Tabulka 9: Dokument Document

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
_id	string	A	N	Null	autogenerate UUID	id tagu, primární klíč
_rev	string	N	N	Null	autogenerate rev. nubmer	číslo revize tagu
couchrest-type	string	N	A	Null		typ dokumentu
name	string	N	A	Null		jméno tagu
description	string	N	A	Null		popis tagu
pdfs	array	N	A	Null		pole s id pdf, které byli přiděleny k tomuto tagu
pdfs_no	integer	N	A	0		počet pdf přidělených k tomuto tagu
templates	array	N	A	Null		pole s id šablon, které byli přiděleny k tomuto tagu
templates_no	integer	N	A	0		počet šablon přidělených k tomuto tagu
created_at	string	N	A	Null		čas vytvoření dokumentu
updated_at	string	N	A	Null		čas poslední úpravy dokumentu

Tabulka 10: Dokument Tag

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
name	string	N	A	Null		jméno pole
default_value	string	N	A	Null		výchozí hodnota
original_name	string	N	A	Null		jméno originálního pole v pdf formuláři
validators	array	N	A	Null		pole obsahující údaje o přiřazených validátorech
type	string	N	A	Null		typ pole
description	string	N	A	Null		popis pole

Tabulka 11: Model Field

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
name	string	N	A	Null		jméno kalkulace
calculation	string	N	A	Null		kalkulace
prePopulate	array	N	A	Null		pole terminálů z kalkulace

Tabulka 12: Model Calculation

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
name	string	N	A	Null		jméno skupiny
fields	array	N	A	Null		pole jmen fildu patřící do skupiny

Tabulka 13: Model Group

Atribut	Dat.typ (cast as)	Klíč	Null	Výchozí	Extra	Český popis
klass	string	N	A	Null		třída reprezentující validator
message	string	N	A	Null		zpráva validatóru

Tabulka 14: Model Validator

C CD-ROM

Na přiloženém CD-ROM jsou uloženy zdrojové kódy PDF portálu. Zároveň CD-ROM obsahuje programátorskou, uživatelskou a administrátorskou dokumentaci. Uživatelskou i administrátorskou dokumentaci spustíte pomocí HTML souboru v jednotlivých adresářích adresáře „prirucky/“.